Data Compression Using Adaptive Transform Coding

by

Martin C. Rost

A DISSERTATION

Presented to the Faculty of

The Graduate College in the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Engineering (Electrical Engineering)

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

October, 1988

This work was supported by the NASA Goddard Space Flight Center under grant NAG-916.

ACKNOWLEDGEMENTS

I thank all the members of my committee for their participation and helpful comments. I am grateful to my readers, Dean Stanley Liberty and Dr. William Brogan, for their suggestions. Their help has improved the quality of this dissertation. I thank Dr. Jerry Gibson for his comments concerning the presentation of the text and for the time he took from his schedule to come and serve on my committee. Special thanks go to my advisor, Dr. Khalid Sayood. It was his patient guidance that kept this ship afloat.

Data Compression Using Adaptive Transform Coding

Martin Christopher Rost, Ph.D.

University of Nebraska, 1988

Adviser: Khalid Sayood

The data of natural images is not stationary, and the coding complexity of images varies from region to region. How well any particular source coding system works is dependent upon its design assumptions and how well these assumptions match the data. In this dissertation adaptive low-rate source coders are developed. These coders adapt by adjusting the complexity of the coder to match the local coding difficulty of the image. This is accomplished by using a threshold driven maximum distortion criterion to select the specific coder used. The different coders are built using variable blocksized transform techniques, and the threshold criterion selects small transform blocks to code the more difficult regions and larger blocks to code the less complex regions.

The different coders are interconnected using a quad tree structure. The algorithm that generates the tree and tests the thresholds is independent of the design of block coders. This allows the system designer to select different coders for the different types of images being coded without having to change the entire coding system.

A progressive transmission scheme based upon these coders is also developed. Progressive coding provides a recognizable image to the viewer in a very short time. As this image is viewed, more data can be received to update the image and improve its visual quality.

A set of example systems are constructed to test the feasibility of these systems of source coders. These systems use scalar quantized and vector quantized transform block coders. Some of the systems are extended to include a new modified block truncation coding scheme. They are used to code both monochromatic and color images with good results to rates as low as 0.3 bits/pel.

A theoretical framework is constructed from which the study of these coders can be explored, and an algorithm for selecting the optimal bit allocation for the quantization of transform coefficients is developed. The bit allocation algorithm is more fully developed, and can be used to achieve more accurate bit assignments than the algorithms currently used in the literature. Some upper and lower bounds for the bit-allocation distortion-rate function are developed. An obtainable distortion-rate function is developed for a particular scalar quantizer mixing method that can be used to code transform coefficients at any rate is also presented.

ACKNOWLEDGEMENTS

I thank all the members of my committee for their participation and helpful comments. I am grateful to my readers, Dean Stanley Liberty and Dr. William Brogan, for their suggestions. Their help has improved the quality of this dissertation. I thank Dr. Jerry Gibson for his comments concerning the presentation of the text and for the time he took from his schedule to come and serve on my committee. Special thanks go to my advisor, Dr. Khalid Sayood. It was his patient guidance that kept this ship afloat.

This work was supported by the NASA Goddard Space Flight Center under grant NAG-916.

Data Compression Using Adaptive Transform Coding

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BTC	block truncation coding
BW	black and white
CRT	cathode-ray tube
DCT	discrete cosine transform
KLT	Karhunen-Loeve transform
LBG	Linde, Buzo and Gray (a type of vector quantizer)
MSE (mse)	mean square error
MBC	mixture block coding
OLSQ	optimal laplacian scalar quantizer
PSNR	peak signal-to-noise ratio
PT	progressive transmission
RGB	red-green-blue
SNR	signal-to-noise ratio
SQ	scalar quantizer
USQ	uniform scalar quantizer
YIQ	luminance/chrominance
VQ	vector quantizer

Chapter 1.

Introduction

Natural image data is not stationary. A low-rate source coder that does not take this into consideration is, in general, not optimal. One type of optimal coder is obtained when the best coding rates are obtained for the least coding cost. The meaning of the terms "best rate" and "least cost" change from application to application. When the overall coding goal is to send single images through a channel whose capacity is substantially less than is required for perfect reconstruction, obtaining a low coding rate is very important. In this case, the image must be coded with some distortion. This distortion is an important coding cost and it must be taken in trade for the decrease in coding rate.

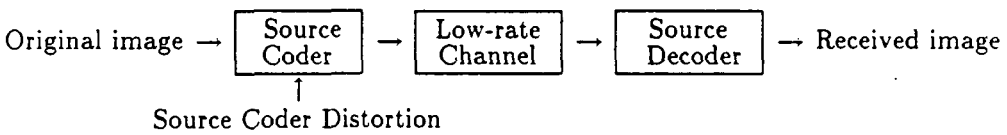


Image data distortion is based upon how the eye sees the data. Visual distortion is difficult to define. Generally, the better the distortion model is, the more difficult it is to use. This, in turn, makes distortion-dependent optimality hard to define. Frequently, the system designer must accept a *good* source coder instead of an *optimal* source coder. Many times, it is better to take an imperfect source coder that "works" than to wait in the hope of finding the best of all possible coders. This is the reality of design engineering. The design of a good source coder for low-rate image transmission is the topic of this dissertation.

When designing a good source coder, many times, the first thing that is done is to use some a priori assumptions that simplify the design process. If an image is coded using the

stationarity assumption, all regions of all images are coded in the same way. This is done without regard for how well, or how poorly, any particular region is coded. The quality of *busy*, or difficult to code, regions is sacrificed only to overcode the non-busy regions. This is undesirable when one is trying to use channel capacity as efficiently as is possible.

In this dissertation, a low-rate adaptive-transform image source coder that takes into consideration the spatial non-stationarities of digitally-recorded images is developed. This adaptive technique allows one to expend more effort coding the difficult regions, and less effort coding the easy regions. This is done by adjusting the number of transform blocks used to code an image region to match its coding complexity. The number of blocks assigned to a particular region is selected through a maximum-distortion threshold criterion. The method is applied to both monochromatic and color images.

The material of this dissertation is organized in two sections: background and new research. The background section, found in Chapters 2 through 4, contains material to introduce the relevant areas of image source coding. In Chapter 2, some of the important features of the human visual system, and the effects of cathode-ray tube display devices are discussed. In Chapter 3, the basics of data compression, scalar quantization, vector quantization, transform coding and block truncation coding are introduced. In Chapter 4, progressive transmission is introduced and some of the methods available in the literature are discussed. The informed reader may which to pass over the background material of these three chapters.

The major results of this dissertation are presented in Chapters 5 through 7. In Chapter 5, the source coding methods of this dissertation, mixture block coding and mixture block coding with progressive transmission, are developed. Computer simulations are given to demonstrate the feasibility of these coders. The details of the distortion measures and the various quantizers used in these examples are also given. A new block truncation coding method is developed and incorporated into the mixture block coding examples. Since these source coders are variable-rate

coders the average coding rate will vary from image to image. Expression are developed for each type of coder to show how the average coding rate is computed. These calculation are a function of the number, size and bit allocations assigned to the different types of coding blocks.

In Chapters 6, distortion-rate functions are developed to model the performance of scalar quantized transform source coders. The standard optimal bit allocation method developed by Huang and Schlutheiss [70] is used as a starting point for this chapter. Once their method has been discussed, it is modified to incorporate the realities of actual quantizers. Also, several upper and lower bounds for distortion performance are computed using the results of this chapter. An obtainable distortion-rate function is also developed. These functions are demonstrated with examples.

In Chapter 7, a set of distortion-rate functions is developed that can be used to study the optimal transform coefficient bit assignment problem as it applies to adaptive transform coding systems found in Chapter 5. This is done by expanding the results of Chapter 6 to include distortion-rate models for vector quantizers and zonally assigned quantization regions.

Finally, in Chapter 8 a review of the significant results of this dissertation are summarized.

Chapter 2.

The Coding and Perception of Images

In this and the next two chapters background material is presented. As an aid to this material, a generic mapping diagram is used. This diagram is used to present the major processes that are involved in the coding of images when using transform methods. This diagram is used to represent the overall coding problem as it applies to the material of this dissertation. Each segment of the diagram that applies to image perception and low-bandwidth image coding are discussed in this chapter. Then, in Chapter 3, the details of the source coding problem as it relates to quantization and block coding methods that are important to work done in later chapters is presented. In Chapter 4, background material concerned with the progressive transmission of images is discussed.

2.1 Modelling Overview

The more that is known about how distortion is perceived, the more the facts of perception can be brought into the source coder design to improve its performance. How distortion is measured by the human visual system is not easily modelled. For optimal performance, the compression scheme must include not only the nonlinear effects resulting from the physiology of the eye, but also must take into account the reason why the image is being viewed. The reason for viewing an image can override almost any other consideration and is dependent upon the application at hand. This, at least with the current modelling theories, makes a complete understanding of image data compression mathematically intractable. Luckily, some of the other important physical features of vision can be modelled without as much difficulty.

The goal of this material is to present some of the more pertinent factors concerning how distortion is perceived and how it effects the design of a low-rate source coder. Since the

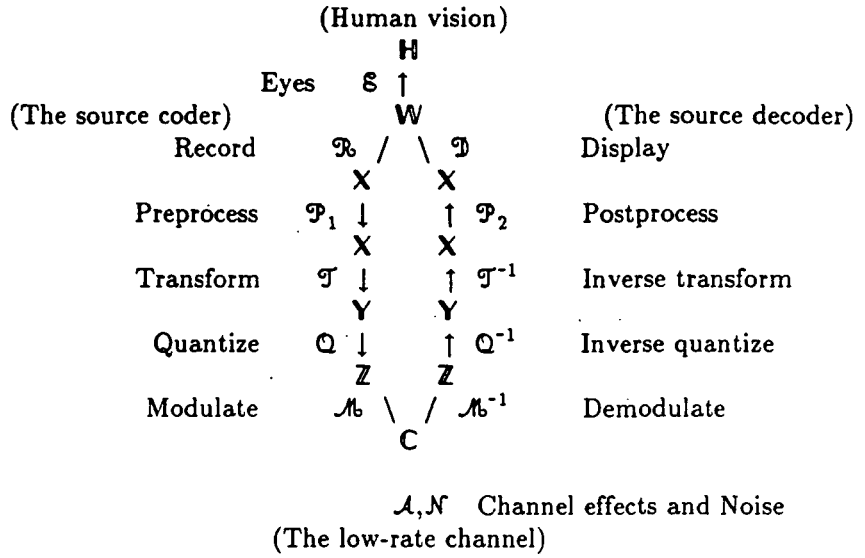
quality of the original data effects the quality of the coded image, a discussion of digital image recording formats is included. The requirements of NTSC television and the database images used to test the algorithms of this dissertation will be introduced. These two facts, how images are perceived and what formats are used to digitally record images, bracket the rest of the source coding system and impose limitations upon how well any given source coder can perform.

Since the type of images studied are for casual observation, such as is typical for television or video-teleconferencing, the aesthetic judgement of the human eye is the only reliable measure of coding distortion. The mathematics of data compression and rate-distortion theory is not a complete indicator of how the eye accepts any given data compression scheme. The human visual system is very difficult to model, and, as a result, much of the design of low-rate source coders of images is a mixture of theory and ad hoc methodology.

In this chapter the source coding of images using transform methods is introduced along with a brief description of what is meant by the term data compression. When designing an image compression scheme several factors must be considered:

- What are the characteristics of the original image data?
- How are images recorded and displayed?
- How is the image perceived?
- How might this be used in a compression algorithm?
- What are the overall data compression goals?
- How much and what kind of distortion, if any, can be tolerated?

The answers to these questions reflect upon the specifics of any source coding method, whether or not the coding involves transform methods. These answers effect the theoretical aspects of how the source is modelled and how the final design is implemented. How these questions apply to the work of this dissertation are introduced through the mapping diagram found below.



As can be seen, the source coding problem can be divided into the following set of processes:

- the source data acquisition, display and visual processes ($\mathcal{R}$, $\mathcal{D}$ and $\mathcal{S}$),
- the source preprocessing and postprocessing processes ($\mathcal{P}_1$ and $\mathcal{P}_2$),
- the compression and decompression processes ($\mathcal{T}$ and $\mathcal{Q}$) and
- the channel processes ($\mathcal{M}$ and $\mathcal{N}$).

The script letters are names used to describe the collection of all possible transformations (mappings) that can be applied to perform the required functions of the particular coding process. In this diagram, there are seven spatial-temporal domains:

H	the human vision system,
W	the "real" world, the image source,
X	the digital image representations,
Y	the transform representations domain of the digital images,
Z	the quantizer and channel symbol indexes and
C	the channel, the medium for data communication and storage.

Each domain is transformed to the next using a mapping from one of the collections mentioned

above. Some of these mappings indicate how humans interpret the physical world. Some represent transformations that can be used by the compression algorithm designer to meet limitations set upon the designer by the physical world.

For example, the collection of eye mappings, $\mathcal{E}$, is all maps used to take world scenes and present them to the observer, $\mathcal{E}:\mathbb{W}\rightarrow\mathbb{H}$. A map $E\epsilon\mathcal{E}$ represents the way a single person might perceive the world. A complication that is found in the mappings of $\mathcal{E}$ is that every person sees the world differently from other people and the way a given person sees the world varies with the application under consideration.

$\mathcal{R}:\mathbb{W}\rightarrow\mathbb{X}$ and $\mathcal{D}:\mathbb{X}\rightarrow\mathbb{W}$ are the collections of all methods for recording and displaying world scenes digitally. More is said about this later.

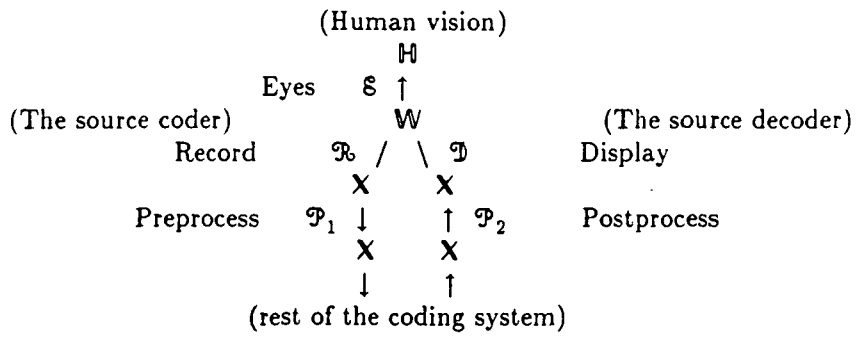
$\mathcal{P}_1:\mathbb{X}\rightarrow\mathbb{X}$ is the collection of preprocessing maps used to condition the recorded images. $\mathcal{P}_2:\mathbb{X}\rightarrow\mathbb{X}$ are the postprocessing mappings used to display the final reconstructed images. These maps are helpful in the reduction of distortion perceived by the final observers by using methods such as dithering, spectrum shaping or coder postfiltering (e.g., [1-4]). A point of notation and nomenclature should be noted. Many times, preprocessing (or prefiltering [1]) is applied to the image before it is digitally recorded to avoid aliasing effects. To avoid confusion, it is assumed that this type of preprocessing is included in the $\mathcal{R}$ mappings. $\mathcal{P}_1$ is applied after the image has been digitally acquired.

The mappings $\mathcal{T}:\mathbb{X}\rightarrow\mathbb{Y}$ transform the images into a new representation that, for one reason or another, is easier to code than the original image itself. Examples of such transforms are the Karhunen-Loeve transform, the fourier transform and the discrete cosine transform. The quantizer maps, $\mathcal{Q}:\mathbb{Y}\rightarrow\mathbb{Z}$, are used to determine the final rate and they induce most of the source-coding distortion found within a given implementation. The quantizer maps the source space (in this case, $\mathbb{Y}$) back into itself and generates a set of indexes that are used to drive the channel modulator. Examples include the Max scalar quantizer [71] and Linde-Buzo-Gray

(LBG) vector quantizers [35]. The transformation/quantizer pair are the parts of the coding process that are studied with greatest interest in this dissertation.

The collections of channel modulation and demodulation mappings, $\mathcal{M}_b:\mathbb{Z}\rightarrow\mathbb{C}$ and $\mathcal{M}_b^{-1}:\mathbb{C}\rightarrow\mathbb{Z}$, and the statistical channel noise mappings, $\mathcal{N}:\mathbb{C}\rightarrow\mathbb{C}$, and the channel media effects (e.g., loss of signal power, scattering, etc.), $\mathcal{A}:\mathbb{C}\rightarrow\mathbb{C}$, determine the integrity of the transmission medium. The nature of the channel itself is the reason why images need to be compression coded in the first place. Most channels have a bandwidth that is surpassed by the large bandwidth requirements of typical digital images. The image bandwidth must be reduced to allow the coding signal to "fit" within the available channel allotment. The presence of channel noise and other forms of signal degradation that are found in the channel complicate the situation. Most coding systems that severely compress data are also more susceptible to noise than those which do not compress the data as much. The "noise" that is of primary concern for the work of the dissertation is that which is incurred in the source coding process. Noise found in the channel will not be studied further, but it is important to notice that it does exist and can harm the quality of the reconstructed images generated by the source coder. Sometimes to a greater effect than the source coder noise.

The following diagram taken from the more complete diagram discussed above presents the parts of the coding system of concern to the remainder of this chapter.



The images of concern are most likely to be transmitted through a digital medium and are

converted for display upon a CRT (cathode-ray tube). The CRT represents the most common display device that is used for television and television-like (e.g., a tele-conferencing) display. Digital communication systems are replacing the analog methods that were originally used for the transmission of television signals. Unless the CRT is replaced by some other display device, its use with these growing digital communication networks set the stage for the future of image transmission. There is need to better understand how these two ideas can be more efficiently used to transmit data.

For any such system, let $R \varepsilon \mathfrak{R}$ represent a specific digitization process used to record world images, and let $D \varepsilon \mathfrak{D}$ represent the final display process used to convert the digital representations of these images into a CRT display. These displayed image are then examined by any number of observers, $E_i \varepsilon \mathcal{E}$, i=1, 2, 3. . . . Each observer may perceive the images of $\mathbb{W}$ in a different manner.

The first fact to come from this presentation is that a CRT cannot perfectly represent the original image, say $W \varepsilon \mathbb{W}$, no matter how perfect the rest of the coding system is. That is,

$$E_i \circ W \neq E_i \circ D \circ R \circ W$$

because

$$E_i \neq E_i \circ D \circ R$$

The recording and display systems are not perfect. A CRT can never render a perfect reconstruction of world scenes. Problems lie in its ability to reproduce color, scale, intensity range, etc. There is also the problem of displaying 3-dimensional data with a 2-dimensional CRT.

This of course, is an overstatement of the severity of the problem, but it does point the way for further study. Specifically, how well does the light intensity of a CRT match that of the original scene it is displaying, and how does it play against the way the eye perceives light intensity. That is the subject of section 2.3. But, before moving into this material a more detailed look is made at the digital recording processes (mappings) of $\mathfrak{R}$.

2.2 Digital images

The coding and resulting display quality of an image is dependent upon how the image is originally recorded. The digital recording of an image must be made with three considerations. They concern the resolution of the recorded image in the spatial, temporal and intensity domains. The real-time presentation of high-quality moving digital images represents a huge amount of data and these huge data loads are compounded when the images are in color.

The picture elements of a monochromatic digital image should be represented with approximately 8 bits of intensity resolution to get adequate image fidelity [1]. If the number of bits is much less than, say, 5 bits, it is possible to see false contours in the image. The figures that are included in Chapter 5 are printed using a device that has only 5 bits of intensity resolution and the false contours are easily seen. Color data, for the images studied, is recorded with 3 color planes each of which is coded with 8 bits of resolution.

The spatial sampling is dependent upon the type of source data to be recorded and the intended presentation medium. For example, it is recommended in CCIR Rec. 601 [6] for digital coding of Y,R-Y,R-G NTSC analog television that a rate of 216 Mbits/s be used. ISDN application for digital telephones lines have specified channels with a rate of 144 Mbits/s for television transmission. As can be seen, images that require such large data rates can be expensive to transmit. High-quality low-rate data compression allows for more television signals to be transmitted within a given rate channel allocation. This reduces transmission costs and increases the number of video channels available to the consumer.

The spatial display of an image is measured in cycles per degree to take into account the final viewing distance from the CRT device. If the image is viewed too closely then there are fewer pels per degree and the spatial quantization effects can be observed. If the observer viewing distance is far enough from the image, the pels falls into a continuum because their size is smaller than the eye's ability to resolve them.

Color in images can be represented with many different formats. The ones most commonly used are the standard tristimulus Red-Green-Blue (RGB) and the luminance/chrominance formats. Examples of the later are the YIQ [7], Y,R-Y,R-G [6], and YUV [8] formats.

The RGB color format is based upon the natural color perception of the human eye. When studying how color distortion is perceived, the RGB colors are useful, but the RGB colors are highly correlated and better data compression can be achieved using a color format whose components are more statistically independent. Such is the case for the luminance/chrominance formats.

The YIQ colors are nearly orthogonal and they have a distinct advantage over the RGB colors. The Y-component represents the image intensity (luminance), and can be used alone to give a monochrome representation of the original image. The I- and Q-components represent the color (chrominance) of the image. Most of the signal energy lies in the Y-component indicating that it carries most of the picture information. To extract luminance data from an RGB coded image all three colors are needed.

New standards for a high-definition television (HDTV) are currently under consideration. The HDTV standard will allow for the transmission of higher quality color television than is currently available [6]. Even though there are many suggested schemes for implementing the new standard, none have been adopted. Most of them have about twice the horizontal and vertical resolution of standard television. Some of them are compatible with current television technology and some are not. It is felt that the coding methods of this dissertation can be applied to the compression of HDTV signals without difficulty and no particular modification to the base algorithms is needed. Since the inter-pel correlations of HDTV signals are greater than those of current television signals it may be possible to achieve a greater degree of compression for HDTV than is demonstrated in this work.

Not all images are represented using natural color intensities. Much of the early work in image compression was done to limit the amount of channel capacity needed to transmit the nine channels of image data recorded by the Landsat system [11,12]. Here the data was compressed for presentation by methods that are now known under the name of vector quantization. In these original works, data features were grouped together by clustering techniques in categories to represent recognizable ground features of interest. Vector quantization techniques are used in the coding methods of this dissertation. These techniques are explained in Chapters 3 and 7.

2.3 Perceptual considerations

When images are used for entertainment, it is desirable to present the eye with something it can feel comfortable with. A data compression system that codes for good feel and comfort is doomed without a more formal definition of what these words mean. There is a large body of literature attempting to give these concepts a more concrete foundation (e.g., [1,14,15,16].) Several of these concepts are discussed below.

When coding images, the expected distortion, as seen by the eye, is based around two important ideas: the type of images coded and the distortion measure used to assess the difference between the original image and its coded counterpart. The algorithm designer wishes to minimize this difference. More mathematically, the expected distortion is

$$E\{d(X,\overline{X})\} \tag{2-1}$$

where $E\{\cdot,\cdot\}$ is statistical expectation, $X \epsilon \mathbf{X}$ is the original image, $\overline{X} \epsilon \mathbf{X}$ is the coded image and $d(\cdot,\cdot)$ is the distortion measure used. To minimize (1), the correct distortion measure must be known and used correctly, if the optimization process is to mean anything.

The description of the visual distortion in this chapter is divided into two sections. The first is concerned with spatial and spectral models for how the eye perceives the world, and the last with temporal models concerning how one might code images for low-bandwidth (low-rate)

transmission. This last section is included to motivate the need for progressive transmission schemes which are the subject of Chapter 4 and 5.

2.3a Spatial, spectral and luminance considerations

The eye responds to image intensity something like a spatial frequency bandpass filter. This is an advantage that can be used to the design image compression systems. Among the earlier work done to quantify visual perception was that of Mannos and Sakrison [5]. Here a measure of just perceptual intensities for monochromatic sinusoids was studied. It was shown that the eye sees these intensities through a filter (Figure 2.1) whose relative response is

$$A(f) = 2.6(\alpha+\beta)e^{-(\beta f)^{1.1}} \tag{2-2}$$

where α=0.0192, β=0.114 and f is the radial frequency in cycles per degree. Since the eyes have a high-frequency cutoff, small spatial features are not seen. The greater the viewing distance, the less distortion is seen.

Since the eye acts as a bandpass filter, image features falling within the passband are seen with the greatest discrimination for perceived distortion. Any coding technique should take this into consideration. If the coding technique introduces only errors outside of the passband, the image is perceived with less distortion than is actually there. High-frequency noise and slow deviations in the background level that are caused by coding are not necessarily perceived, and these out of band frequency ranges can be coded with less care. Most of the coding effort can be spent processing the visible passband frequencies. The coding methods of this dissertation attempt to offer a solution to the problem of coding images that contain edges and fast gradients. Since edges are important to the aesthetic perception of an image these features must be coded with care. The coding techniques of this dissertation are designed to help improve the coding of these features without overcoding other parts of the image.

The minimum fidelity with which a given object needs to be coded to guarantee subject recognition is strongly dependent upon the type of object being viewed [12]. For example, more distortion can be tolerated in the presentation of individual faces before the face is classified as unrecognizable than is the case for, say, a crowd scene. The change in perceived distortion as the recognition level of the subject matter changes is important to the overall evaluation of image data compression systems. To make use of subjectivity, the coder must have a very high level of object modelling and recognition, and to do so requires a large amount of computation and sophisticated application oriented algorithms. If the tools to perform such tasks are not available, as is usually the case, the images must be coded ahead of time so the coding distortion can be evaluated on a case-by-case basis with human intervention. The coder can be adjusted accordingly for best performance. Either method requires large amounts of pre-presentation processing. If one is designing a database for the random viewing of known sets of images sometimes the overhead can be tolerated. The amount of time spent coding an image is small when divided among the many times it will be recalled for viewing. But these considerations lie outside the scope of the work done here. We are satisfied with distortion models that measure how easily the image data is coded over regions of varying local busyness.

Equal changes in luminance are not perceived equally by the eye. It has been found that "just visual" perception is nearly linear when perceived changes in luminance are normalized by the mean background luminance

$$\Delta L/L = \text{constant} \tag{2-3}$$

This effect is called Weber's law [17]. By defining contrast as

$$C = \ln L \tag{2-4}$$

small changes in constant contrast, $\Delta C = \Delta L/L$, are measured along a straight line.

There are several other ways to model contrast, among them are [13]

$$C = a \ln(1+bL) \tag{2-5}$$

where $a=b/\ln(1+b)$, $L\varepsilon[0,1]$ and $a\varepsilon[9,11]$; and [5]

$$C = aL^b \tag{2-6}$$

where a=1 and b=1/3. Any of these contrast functions can be used to weight the distortion measure used in the coding process to reflect more correctly how the eye perceives distortion.

Another effect of importance to the measure of visual perception is the way cathode-ray tubes present the luminance data. One model for CRT intensity shows that it is not linear,

$$L = kE^d \tag{2-7}$$

where E is the applied electric potential, k is a constant, and $d\varepsilon[2,2.5]$ is the illumination exponent [14]. The nonlinearity of luminance as a function of electric potential partially balances the nonlinearities of the Weber effect, since b of (6) and d of (7) tend to offset each other. When the luminance level is low, the situation is slightly more complex because the tube brightness gets washed into the ambient background, for example

$$L = kE^d + L_{amb} \tag{2-8}$$

So it is difficult, if not impossible, to attain a good low level luminance performance measure that is independent of the viewing environment.

Since the CRT is the preferred display device of this work, and the exponential factors of perceived contrast and CRT brightness tend to cancel each other. Henceforth, only absolute differences and mean square differences will be used to measure distortion.

The two models presented above do not take into consideration the effects of changing background luminance and the effects of high-contrast regions that are near those of normal or low luminance. These effects are very nonlinear and will not be considered here. Some of these effects are discussed in [14].

The correct distortion measure for color images is also perceptually nonuniform and is still an open research area [8]. Not only are the intensity considerations introduced above important but it is known that small changes in color are perceived with very nonlinear effects. The

authors of [8,16] suggest some complex nonlinear transformations of the camera RGB outputs. The review paper of Limb, Rubenstein and Thompsom [16] and material found in Pratt [17] are good starting places for a more complete study of these problems.

As an example of the effectiveness of visual distortion weighting, consider a paper by Edgerton and Srinath [16]. They developed a scalar quantizer based upon the visual distortion model of Mannos and Sakrison [5]. They use the Mannos and Sakrison model to select the scalar quantizer bit assignments used to code the coefficients of 16×16 DCT coded image blocks. The quantizer bit assignments are made using an mse distortion criterion that is constrained to give a predetermined entropy coding rate. They use their method to code a single 512×512 image at 0.5 bit/pel. By using the visually weighted distortion criterion they state that their system performed close to that of a 2.0 bit/pel DCT coder designed without the visual weightings.

Another important feature related to the above material is that transform source coders designed with the YIQ color system show about half the mse of equivalent RGB based systems [18]. These results are similar to those obtained by the computer simulations discussed in Chapter 5.

The material introduced above can be utilized in the compression of images. Brightness-to-contrast scaling can be implemented as both a pre- and post-processing function. While frequency weighted compensation is best done as part of a transform data-compression step where the weighting factors can be directly implemented in the source coder design [19]. Some nonlinear postprocessing functions can be used to remove the undesirable characteristics of the source coder. An example is presented in [20].

2.3b Low-bandwidth temporal considerations

The nontemporal vision models considered in the last section have been extended to include temporal effects [21,22]. These models are complex, and even when simplified [23], do

not readily lend themselves to the present application where still, or nearly still, images are of the most interest. If the distortion models are too complex, they can increase the computational burden of the final implementation to a point where it can be made useless for real-time data processing.

The following discussion is motivated by a need to understand progressive transmission as an alternative to slow scan image methods and understand the basics of how distortion is perceived in the temporal framework of slow moving images. Sometimes the images are updated as rarely as one image every several minutes. The mid- to high-frequency temporal response of the visual system presents some very specific problems that must be addressed. At these low framing rates, the manner in which the CRT image is updated is of primary interest. The temporal models used to describe these effects are more heuristic than the models mentioned above, but they are useful in the results that they show.

When temporal considerations are included, the eye perceives images in a manner which must include how the image is presented for viewing. If, due to bandwidth considerations, the image must be reconstructed slowly, the eye will notice the reconstruction process. These secondary considerations come into the forefront of image perception, and may dominate the entire coding process.

For example, when the image is scanned line-by-line, as is the case for ordinary television, the scanning process is not seen by the eye because the scanning is performed at rate above the temporal-spatial frequency range of the eye. But, in the case where low-bandwidth transmission is used, it may take many seconds to scan a single image. Here the scanning process is clearly seen by the eye. The image is perceived as a still image except where the scan is updated. The update process attracts the eye, and causes the image to be viewed with most of the user's attention being expended watching the scanning process. Therefore, less time is spent studying the image itself. The eye fixation process of slow scan television is the driving

force behind the development of progressive transmission schemes, where the image is reconstructed as a whole on a frame-by-frame basis instead of pel-by-pel. More is said about the specifics of progressive transmission in Chapters 4 and 5.

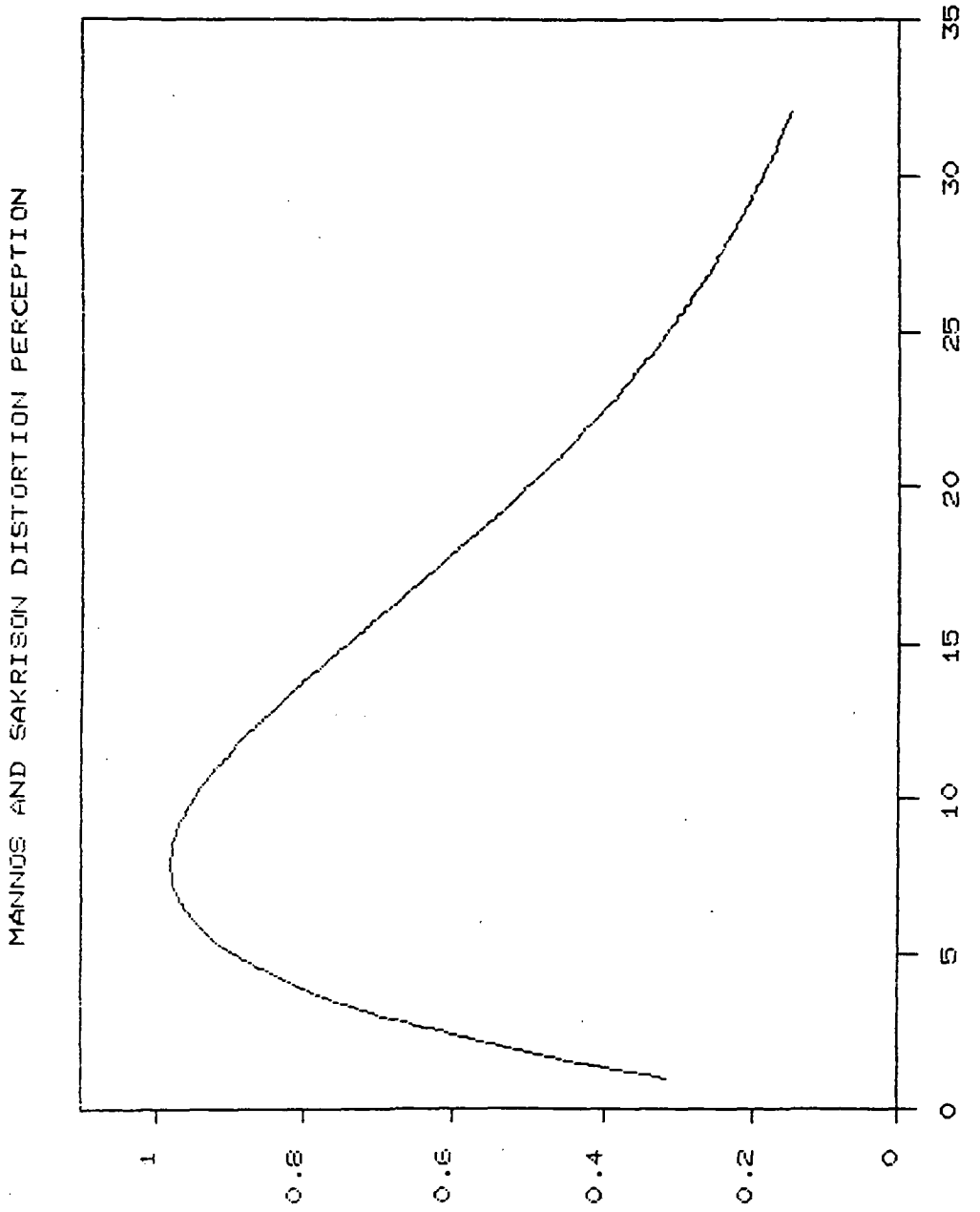


Figure 2.1 Mannos and Sakrison visual perception model

Chapter 3.

Image Compression Techniques

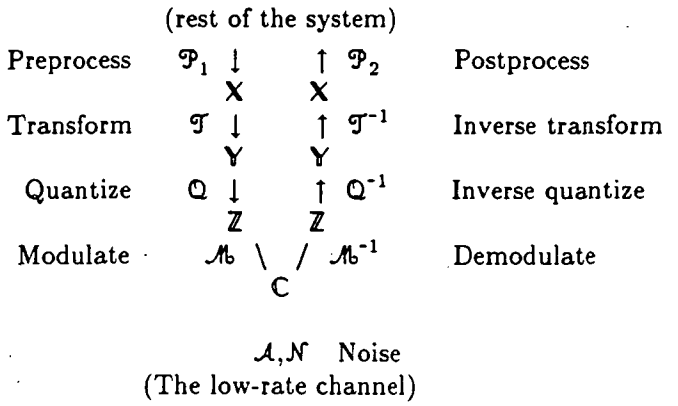
In this chapter a review of the image compression techniques, that will be used in later chapters, is made. Image compression, or more generally, data compression can be thought of from two major points of view. In the first, it is desirable to reduce the amount of data needed to reproduce the original data without any coding losses. Such source coders are called noiseless, information preserving or lossless coders. They are primarily concerned with compressing the original data so that it is completely recoverable. Henceforth, this type of coder will be referred to as a lossy coder. These techniques are useful for minimizing the required channel capacity when there is a need for perfect reconstruction. The need for perfect recovery severely limits the amount of compression that can be achieved. To first order, the amount of compression that can be achieved is based upon the single-symbol entropy of the data. For typical natural images coded with eight bits of resolution the entropy is usually between 5 and 7.5 bits. Since images have large inter-pel correlation, the number of required coding bits can be reduced by using a redundancy or correlation removal coding scheme. Then the best coding rate is predicted by the equivalent multiple-symbol entropy.

The second point of view is important when a higher degree of compression is required than can be achieved with lossless coding. Here the desired rate can be much less than the source entropy. If this is true, some loss in data integrity must be expected. These losses are a direct result of Shannon's coding theorem [41]. This type of source coder is called a lossy coder. Lossy coding relies strongly upon many-to-one mappings to attain high levels of compression. Because these mappings are many-to-one they are not invertible, and perfect reconstruction of the source data is not possible. But, their losses can be controlled by causing the coded data to

remain within some definable distance from the original data using a norm that is defined upon the source and coded data sets. The primary class of lossy mappings used for data compression are the various forms of quantization.

Quantizers can be used in conjunction with other one-to-one (invertible and lossless) mappings, such as transform coding, to decrease perceptual losses and increase compression as much as possible. The goal is to introduce the quantization schemes and transform coding methods used in image data compression coders of later chapters.

Consider the portion of the mapping diagram of Chapter 1 that applies to the material of this chapter:



In the next section the measurement of data compression and image quality is discussed. Then in the following sections quantization and transform coders are introduced. In the last section of this chapter, block truncation coding (BTC), a useful moment preserving coding system is introduced. BTC and transform coding are used together to form one of the mappings of $\mathcal{T}$ that are used in the computer simulation of this dissertation.

3.1 Measuring data compression and image quality

How well a data compression system works is measured using two important yardsticks. The first, the data-reduction ratio or data-compression ratio, is a measure of the reduction in channel capacity needed to transmit a given image. This is a ratio of average coding rates, and

is defined to be

$$\frac{R_{orig}}{R_{coded}} \tag{3-1}$$

where R_{orig} and R_{coded} are the average coding rates before and after the compression coder is used.

In image compression, R_{orig} usually represents the coding rate that is required for transmitting (or storing) the image data using a uniformly weighted symbol set. For the images considered here, this rate is 8 bits/pel (picture element) for monochrome images and 24 bits/pel (8 bits/color) for color images. Let it be assumed that R_{orig} represents the rate obtained for a uniformly coded source. If an entropy coded symbol set is used to code the image data, instead of a uniformly coded symbol set, the average rate can be reduced by some degree that is dependent upon the correlation of the original data.

An entropy coder selects a different coding weight for each source symbol. The coding weight of a symbol is equal to the average number of channel symbols that are used to transmit it. This weighting is inversely proportional to the symbol's probability of occurrence when using the Shannon entropy function. The average coding weight of the i-th source symbol occurring with probability p_i is

$$r_i = -\log_2 p_i \tag{3-2}$$

The average entropy rate for the monochrome image symbol set (or source alphabet), that has 256 (2^8) elements, is

$$R_{ent} = -\sum_{i=1}^{256} p_i r_i = -\sum_{i=1}^{256} p_i \log_2 p_i \tag{3-3}$$

As it is defined here, R_{ent} is known as the single-symbol source entropy.

In general, the individual symbol rates of R_{ent} are not integers and a single-symbol entropy code (such as the Huffman code [42]) does not achieve this rate. In reality, R_{ent} is a lower bound for any single-symbol entropy coder [74]

$$R_{ent} \leq R_{coded} \leq R_{orig} \tag{3-4}$$

The last of these inequalities tells us that an entropy coder is, in itself, a data compression coder since

$$R_{coded} \leq R_{orig} \tag{3-5}$$

Even though the coders designed for this dissertation do not use entropy coding, entropy coding is useful because it offers a theoretical lower bound against which any coder can be compared. Therefore, all lower bounds will be calculated assuming uniform codeword lengths.

Since most of the literature concerning image compression uses monochromatic source images that are coded with 8 bits per pel (8 bits per color for color images) the compression ratio is generally simplified to just the compression rate R_{coded}. This convention is also used here. So an image that is coded with an average rate of 0.5 bits/pel has a compression ratio of 16.

The second yardstick used to measure the performance of a data compression system is concerned with data integrity. Here, all of the compression coders of interest compress the source image to such a low coding rate that the original image cannot be recovered without some loss of information. For a given coding rate, it is desirable to limit the coding losses, in the hope that the lossy reconstructed image "looks" as much like the source image as is possible. The quality of a coded image is measured in many ways, but mean square error (mse) is one of the more convenient measures of distortion and it is used to measure quantitative image quality in this dissertation.

With the mse measure, the ratio of error variance, σ_q^2, to source variance, σ_s^2, is used to measure reconstructed image quality. Two factors effect this ratio

$$\frac{\sigma_q^2}{\sigma_s^2} = \epsilon^2\gamma^2 = G \tag{3-6}$$

where ϵ^2 is the performance factor of the quantization system, and γ^2, the spectral flatness measure (sfm), indicates how well the transformation mapping "whitens" the source. The flatter the source spectrum can be made the more compression can be achieved. Sometimes these two factors are not separable, so they must be thought of as a single more general factor:

the coder compression or gain factor, G. This is discussed in more detail in Chapters 3, 6 and 7.

It is often convenient to modify (6) using a log-ratio function

$$\text{SNR} = -10 \log_{10}\frac{\sigma_q^2}{\sigma_s^2} = -10 \log_{10}\epsilon^2\gamma^2 = -10 \log_{10}G \tag{3-7}$$

called the signal-to-noise ratio (SNR) of the source coder. SNR measures coding loss in decibels. The logarithm allows the coder performance to be measured as a sum of the quantizer performance and the transformation process

$$\text{SNR} = -10 \log_{10}\epsilon^2 - 10 \log_{10}\gamma^2 = \text{SNR}_q + \text{SNR}_t \tag{3-8}$$

While this formulation is a simplification for how any source coder might induce distortion, it does offer a distinct division of coding effort that matches the way source coders are usually designed. In this dissertation, the transformation is selected a priori and the quantization processes are designed with the transformation as a design constraint.

In image coding, another important point is concerned with how the signal-to-noise ratio is measured. Reconsider equation (7), here the SNR is dependent upon the variance of the image coded. When comparing *equal* distortion levels for two different images their signal variances and signal energies will, in general, be different. This causes their SNR values to be different. This can destroy the usefulness of SNR values for comparing different coders. To overcome this problem the definition of SNR is modified

$$\text{PSNR} = -10 \log_{10}\frac{\sigma_q^2}{\sigma_{white}^2} = -10 \log_{10}\frac{G\sigma_s^2}{\sigma_{white}^2} = \text{SNR} - 10 \log_2\frac{\sigma_s^2}{\sigma_{white}^2} \tag{3-9}$$

to use a standard definition of signal variance: where σ_{white}^2 is the white reference energy level of the image. When using 8-bit pels, $\sigma_{white}^2=255^2$. The white-referenced SNR is called the peak SNR (PSNR) and it is a direct measure of coding distortion that is not effected by the image variance or dc background level. Henceforth, PSNR is the measure of coding distortion used.

3.1 Quantization

Quantization is a classification process whereby the elements of a large, possibly infinite, set of elements are identified with the elements of a smaller finite set. The larger set, specifically a collection of possible images representations, $\{X\} \subseteq \mathbf{Y}$, is known as the source and the smaller set of representative symbols used to code the images, Y, is known as the quantizer code book. The elements of Y are imbedded into the source space and the classification process is usually a nearest neighbor mapping. The source points are mapped to a nearest neighbor taken from the code book by using some preassigned distortion measure. That is, the source elements, $x \in X$, are matched to the elements of $Y=\{y_i | i \in \mathfrak{I}\}$ by the many-to-one mapping

$$Q(x) = \min_{\{y_i\}} d(x,y_i) \tag{3-9}$$

where d is the distortion measure. The quantizer index set, $\mathfrak{I} \in \mathbb{Z}$, is mapped into a set of channel symbols, Ψ, by some modulation technique, $M \in \mathcal{M}$,

$$M(\mathfrak{I}) \rightarrow \Psi \tag{3-10}$$

and the channel symbols are demodulated at the receiver

$$M^{-1}(M(\mathfrak{I})) \rightarrow \mathfrak{I} \tag{3-11}$$

The receiver decodes the indicated index representative, $i \in \mathfrak{I}$, from the received channel symbol, to recover the original quantized source value, $i \rightarrow y_i$. The set of quantizer values, $\{y_i\}$, must be known to the receiver a priori, or the receiver must be able to be reconstruct them from side information provided by the transmitter.

An important point that should be noticed is that the size of the channel symbol set (let the channel index set be $\mathfrak{K} \in \mathbb{Z}$ where the channel symbols are $\Psi=\{\psi_k | k \in \mathfrak{K}\}$, and let its size be $|\mathfrak{K}|=|\Psi|$), is not always the same as the size of the quantizer code book

$$|\mathfrak{K}| \neq |\mathfrak{I}| \tag{3-12}$$

The lack of equality in the size of these sets indicates that the modulation mapping is not necessarily one-to-one. For example, a block (or vector) of n quantizer indexes may be coded

into a single channel symbol

$$\phi_k = \mathrm{M}(i_{k1}, i_{k1}, \ldots, i_{kn}) \text{ where } i_{kj} \varepsilon \mathfrak{I} \tag{3-13}$$

Therefore, the number of code book elements and the number of channel symbols is not the same. Henceforth, let the set of quantizer code book elements be known as the reproducing alphabet, and the set of channel symbols be known as the channel alphabet. Clearly, the modulation mapping determines the coding rate of the quantizer output in the channel.

The channel rate is based upon whether the elements of $\mathfrak{I}$ (or $\mathfrak{K}$) are uniformly coded or are entropy coded (e.g., Huffman coding [42]). Entropy coding generally uses fewer coding bits than uniform coding, this is an advantage when designing a low-rate system. But, if there is noise in the symbol coding/decoding process, entropy coding usually suffers more signal degradation than uniform coding.

With entropy coding the code book symbol lengths are different, and noise will cause the decoder to loose synchronization with the source coder. Once the decoder has lost sync it may never resynchronize correctly (e.g., [80]). Since all the symbols of a uniform coder are of the same length it is more difficult to loose synchronization. This ability to remain synchronized in the presence of noise is an inherent advantage of uniform coders over entropy coders.

Since most real data transmission systems have some noise, the tradeoff between entropy and uniform coding must be weighed against one another when designing a data compression system. But, the tradeoffs between source and channel coding fall within the realm of joint source/channel coding theory and are outside the subject of this dissertation. Therefore, all of the computer simulations of this dissertation use uniform coding. The gains that can be incurred through the use of entropy coding are well known and the specifics of how they can be used for the source coding methods developed in later chapter is left for future study.

Since perceptual losses are hard to model, the actual distortion measure used to build image compression schemes are usually simplifications of these difficult models that are easier to

work with (e.g., mean square error or maximum absolute error). The loss of accurate models is taken in trade for mathematical tractability. There is a large body of literature that has studied the problem of mean square error. The Max quantizers mentioned above are optimal quantizers for the mse distortion criterion.

All source points mapped into a single code book element are known collectively as a quantization region. The shape of the quantization regions are a function of the distortion measure used, and their size determines the average distortion of the quantization process

$$d_Q = \frac{1}{|\mathcal{Y}|} \sum_{\{i \in \mathcal{Y}\}} \int_{V_i} p(x) d(x, y_i) dx \tag{3-14}$$

where the p(x) is the probability density function of the source and the V_i are the quantization regions assigned by the distortion measure d for a quantizer Q of size $|\mathcal{Y}|$. If mse is the distortion measure used, the average distortion is

$$d_Q = \frac{1}{|\mathcal{Y}|} \sum_{\{i \in \mathcal{Y}\}} \int_{V_i} p(x)(x - y_i)^2 dx \tag{3-15}$$

and the quantizer performance can be measured using decibels

$$SNR_q = -10 \log_{10}(d_Q / \sigma_s^2) \tag{3-16}$$

The quantization regions and the spatial placement of the code book elements are usually chosen to minimize the average distortion incurred by a given quantizer. More is said about this later.

Since the code book can have substantially fewer elements than the source data set, compression can be achieved. The compression ratio obtained is related roughly to the logarithm of the ratio of the source size to code book size. (This is true because the number of bits needed to code such a set is related to the logarithm of the set size.) The inherent and universal ability of quantizers to compress data by a many-to-one mapping makes them a valuable tool to the compression algorithm designer.

Sometimes the quantization regions or the placement of code book elements are constrained in some manner. Usually, the constrains are added to speed up the quantization pro-

cess by simplifying the quantizer hardware and computational algorithm. If enough constraints are placed upon the quantizer it is possible to completely predetermine the quantizer code book without using the underlying distortion measure or source characteristics. The more constraints placed upon the quantizer the less optimal its performance can be. Therefore, there is a tradeoff to be made between the ease of using a quantizer and the coding quality it can attain. The computer simulations of Chapter 5 include results showing these effects. With this overview in mind, several types of quantizers are now discussed.

3.1a Scalar quantization

The code book elements of a scalar quantizer are points placed in a one-dimensional space, and their quantization regions are line intervals. The average distortion for a scalar quantizer is

$$d_Q = \frac{1}{n} \sum_{i=1}^{n} \int_{a_i}^{a_{i+1}} p(x) d(x, y_i) dx \tag{3-17}$$

where n is the number of code book elements and $(a_i, a_{i+1}]$ is the i-th quantization interval containing quantizer element y_i. Max [71] developed an iterative algorithm to optimize the placement of the interval end points and code book elements for a given code book size, source pdf and distortion measure. The algorithm alternates between calculating new end points given a fixed code book, and calculating new code book elements given fixed end points.

If the code points and the interval end points are allowed to find their own placement without constraints, the quantizer is called optimal. If the code book elements or end points are uniformly spaced (e.g., $a = na_0 + b$, where $n \in Z$ and $a_0 \in R^+$), the quantizer is called an optimal uniform quantizer. The Max scalar quantizers have been published for several pdfs (Gaussian [71], Laplacian [73] and gamma [75]). The optimal laplacian quantizer is used in the computer simulations of Chapter 5.

The entropy coding rate for a scalar quantizer is

$$R = -\sum_{i=1}^{n} \int_{a_i}^{a_{i+1}} p(x)\log_2 p(x)dx \leq \log_2 n \tag{3-18}$$

and $\log_2 n$ is the uniform coding rate. It is assumed the symbols are coded using binary representations.

3.1b Vector and lattice quantization

Vector quantization is a natural extension of scalar quantization into m-dimensional spaces. The quantization regions are m-dimensional nearest neighbor polytopes. The average distortion is found by integrating the source pdf over each of these polytopes, V_i,

$$d_Q = \frac{1}{n} \sum_{i=1}^{n} \int_{V_i} p(x)d(x,y_i)dx \tag{3-19}$$

where the x and y_i are m-dimensional points from the source space and, similarly, p(x) and $d(x,y_i)$ are the m-dimensional source pdf and distortion functions. The entropy rate is

$$R = -\sum_{i=1}^{n} \int_{V_i} p(x)\log_2 p(x)dx \leq \log_2 n \tag{3-20}$$

If vector quantizers did not offer performance improvements over their scalar counterparts there would be little cause to use them. Vector quantization can be computationally more intensive than scalar quantization, especially for optimal quantizers assigned to nonuniformly distributed sources. However, vector quantizers do give better distortion performances than the best scalar quantizers. If the source model and distortion measure are simple to model, it has been shown [30] that a vector quantizer can, with certain assumptions, asymptotically achieve the rate-distortion limit. The Shannon lower bound can be used to predict the best performance of any quantizing system for a given source [74]. The Shannon lower bound is used in Chapters 6 and 7 to compute lower bounds for different quantizers of transform coefficients.

There are many ways to design vector quantizers. Three methods are mentioned below. The first method is the (locally) optimal LBG method. The second method uses mathematically modelled pdf's, and the third using lattice structures to place the quantizer points.

The LBG method builds a code book using a finite set of training vectors. and it can be implemented without any knowledge of the source statistics, since it minimizes distortion over the training set only. The final code book is not guaranteed to be unique or globally optimal. One of the drawbacks of the LBG method is that it may not perform well when used to code source points not found in the training test. This causes problems when it is difficult to build an adequate training set.

The design of an LBG quantizer is an iterative technique. It alternates between placing the code book elements assuming fixed quantization regions, and finding the quantization regions assuming a fixed code book. The quantization regions are disjoint partitions of the training set.

The algorithm is computationally intensive, especially when large training sets are used or large code books are being generated. To help overcome this problem, one can start with a small code book. Once the small code book is optimized, its size can be increased using a point splitting technique [35], and the LBG minimization algorithm is repeated. The code books are divided and reoptimized until the desired code book size is attained.

The split-and-optimize method can be computationally tedious. It is expensive to design a full-blown code book for each and every step. It is pointed out by Hang and Haskell [44] that the computational load of LBG code book design can be eased by using the nearest neighbor (NN) algorithm of Equitz [36] to design the preliminary code books, and using the LBG algorithm for only the final stages of the design. Using their method a computational load improvement of 20:1 was reported.

Other than the computational load problems of code book design, the LBG method has another problem. The code book is not guaranteed to be unique. A poor choice for the initial code book may result in a poorly constructed final code book. It is possible to arrive at significantly different final code books by using different starting sets.

Once the code book is designed, using it to code source vectors can also be computationally intensive, since LBG code books have no structural regularity that can be used for fast coding. Paliwal and Ramasubramanian [26] have suggested a method where the code book vectors are sorted by their expected hit probability to improve the coding time, and they also suggest an early exit method to stop unnecessary computing when calculating the distortion level for each code book vector. But, even with these improvements the coding process is still time consuming.

When considering an LBG design, its disadvantages must be weighed against the expected improvement in distortion performance that can be obtained. An alternative to the LBG vector quantizer is to use a more constrained design, such as can be obtained from the uniformly placed code book points of a lattice quantizer [27,32,33,34].

The code book elements of a lattice quantizer are made from multiples of a set of basis vectors $\{e_j\}$. A lattice code point is defined to be

$$y_i = \sum_j \alpha_{ij} e_j \tag{3-21}$$

where the α_{ij} are integers. If the α_{ij} and x_j are properly selected, source points can be more quickly coded than those of an LBG code book. Some lattices quantizers can be coded at speeds proportional to the dimension of the source space. Such coding speeds are similar to that obtained when using a scalar quantizer.

Another approach is only useful when the source has a simple pdf whose n-dimensional characteristics can be used to simplify code book design such as with the uniform pdf [28,41]. Fischer [37] develops a quantizer using the properties of the laplacian pdf. The quantizer code

points are placed on the pyramidal surface that is defined by the mean of an independent identically distributed source. For large dimensions, Fischer's quantizer seems to approach the rate-distortion bound, and he offers a fast coding method to improve its coding speed. This quantizer may be useful as an alternative to the LBG quantizers used in Chapter 5, if it were desirable to improve the coding speed.

3.2 Transform coding

Distortion-rate theory leads one to the conclusion that if one can classify the ensemble of all images with a probability measure then the ensemble can be coded in some way to meet the distortion-rate bound [74]. But, two problems immediately come to mind when one tries to implement this theory. Firstly, it seems impossible to find a tractable model for the image ensemble. The possible number of, say, 512×512 images is incredibly large ($2^{8 \cdot 512 \cdot 512}$). It can be very difficult to adequately model ensembles of this size. Secondly, distortion-rate theory gives no useful information as to how one might design a coder that attains the bound it sets forth. Most systems that do approach the distortion-rate bound, approach it by requiring a immense amount of coding effort. Some simplifying scheme must be developed.

Block coding is one way to reduce the ensemble size, but even for small blocks, say 8×8, the ensemble is still very large ($2^{8 \cdot 8 \cdot 8}$). There is little conformity among the members of the ensemble and most compression schemes require that the source have very definite and very amenable characteristics. Therefore, the problem is still hard to solve.

But, it is known that the pels (picture elements) of small blocks are highly correlated for natural images. The high inter-pel correlation is where most spatial block coding techniques acquire their ability to compress image data. Also, as one would expect, this is where transform coding methods offer help. The correlation phenomenon tends to push most of the block energy into a small number of the transform coefficients, and these coefficients tend to be more statistically independent than their spatial domain counterparts. Most of the image can be

coded using a few, specifically identifiable, transform coefficients. Since the transform coefficients are less dependent, more information per coded bit can be attained [25]. Additionally, some of the effects of perceptual errors are more easily modelled in the transform domain. In [16], as was mentioned in Chapter 2, a spectral model of human vision [5] is used to select a more judicious DCT coefficient quantizer bit weightings.

An upper bound for the expected mse coding error of a transform source coder, σ_q^2, can be found by using the spectral flatness measure [1]

$$\sigma_q^2 \geq \epsilon_q^2 \gamma_{xy}^2 \sigma_s^2 \tag{3-22}$$

where ϵ_q^2 is the quantization performance factor and γ_{xy}^2 is the transform coder spectral flatness measure. The performance factor is a function of the quantizer used and is discussed in Chapters 6 and 7. The spectral flatness factor is defined to be

$$\gamma_{xy}^2 = \exp\left\{\frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int \ln S_{xy}(e^{j\omega_x}, e^{j\omega_y})d\omega_x d\omega_y\right\}/\sigma_s^2 \tag{3-23}$$

where S_{xy} is the source power spectral density (psd) function. The spectral flatness measure satisfies

$$0 \leq \gamma_{xy}^2 \leq 1 \tag{3-24}$$

The flatter the psd, the closer γ_{xy}^2 is to 1, and the greater the resulting source coder distortion. The spectral flatness measure, being the fourier transform of the source autocorrelation function, $R_{xy}(x,y)$, is generally formulated using the assumption of stationary statistics. Even though R_{xy} can be formulated as a nonstationary function, it offers little immediate help to solving the source coder design problem. Instead, it can be simpler to model the mse problem using the gain factor representation

$$\sigma_q^2 = G\sigma_s^2 \tag{3-25}$$

and attempting to minimize G. The minimization of G is the general approach taken for this dissertation.

3.2a KLT and DCT coding

A general two-dimensional unitary (orthogonal) transform is of the form

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^{\mathrm{T}} \tag{3-26}$$

$$\mathbf{X} = \mathbf{A}^{\mathrm{T}}\mathbf{Y}\mathbf{A} \tag{3-27}$$

where $\mathbf{X}$ is the source data matrix, $\mathbf{Y}$ is the matrix of transform coefficients, and $\mathbf{A}$ is the unitary matrix of eigenvectors.

When choosing the $\mathbf{A}$ matrix to be used with a source coder some selection criterion must be used. One criterion selects a transform matrix that places as much energy as possible into a fixed number of the transform coefficients. If such a transform matrix is selected using an optimal mse fit [43], the Karhunen-Loeve transform (KLT) results. For the KLT the $\mathbf{A}$ matrix is built from the eigenvectors of the source image covariance matrix.

If all the images to be coded are similarly stationary, then the $\mathbf{A}$ matrix need only be constructed once and transmitted to the receiver. Then by quantizing the transform coefficient matrix a efficient low-rate representation of the original image can be coded for transmission. If not all of the coefficients are to be coded, due to bandwidth limitations, the coefficient-limited fit is still optimal.

Images are not stationary and the inter-pel covariance matrices of any two images are hardly ever the same. What is usually done in practice is to design a new $\mathbf{A}$ matrix for each image to be coded. Since the receiver does not know what the $\mathbf{A}$ matrix is like for any given image it must also be quantized and transmitted, along with the transform coefficients. The coding overhead incurred in this process causes the KLT not to be a good method for designing low-rate source coders. If a standard transform matrix could be selected that would perform well for typical images, then the coding problems of the KLT method could be overcome, and all of the channel capacity could be used for the transmission of the transform coefficients alone.

The discrete cosine transform [19] (DCT) is such a transform, it uses the same transform matrix for all images and is known to code near the performance level of the KLT for almost all natural images [20]. Like the fourier transform the low-ordered transform coefficients represent low-order spatial frequency components in the original image. This makes the DCT desirable since most natural images tend to carry most of their signal energy in the low-order components. Often the higher frequency components can be ignored when coded for very low data rates. The DCT is used for all of the computer simulations of Chapter 5.

The elements of the $n \times n$ DCT transform matrix are

$$A_{ij} = \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} B_{kl} \cos\left\{\frac{(2k+1)i\pi}{2n}\right\} \cos\left\{\frac{(2l+1)i\pi}{2n}\right\} \tag{3-28}$$

where the constants are

$$B_{00} = 1/n$$

$$B_{0j} = B_{i0} = \sqrt{2}/n \text{ and } B_{ij} = 2/n \quad \forall i,j \neq 0$$

Once the type of transform to be used is selected several other considerations are important [21]:

- the transform blocksize,
- the coefficient truncation zone,
- the quantization types used to code the coefficients,
- the desired coding rate and the bit allocation in the coding zone and
- the final image quality (i.e., mse)

The specifics of how each of these items is used for the computer simulations of the dissertation are discussed in Chapter 5.

Once the transform is selected the problem of choosing which coefficients of the transform block are to be coded and the problem of assigning the number of bits that are used to code each them must be approached. The theory behind the bit assignment of scalar quantized coefficients is the subject of Chapter 6, and the theory of bit assignment for vector

quantized coefficients is the subject of Chapter 7.

3.3 Block truncation coding

It is sometimes desirable to preserve some of the statistical properties of the original image in the reconstructed image. Delp and Mitchell [64, see also 45] developed a nonparametric image coding method, called block truncation coding, used to preserving the first (mean) and second moments of a block of monochromatic image data. Their method preserves the computed mean

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i \tag{3-29}$$

and the second moment

$$\sigma^2 = V^2 - \bar{X}^2 \text{ where } V^2 = \frac{1}{m}\sum_{i=1}^{m} X_i^2 \tag{3-30}$$

of a block, where there are m pels (X_i) per block, by selecting appropriate quantization values for a one bit quantizer. A one bit quantizer output must be at one of two levels, say, a or b. The quantization rule used by Delp and Mitchell set the quantized value, X_i', of X_i by comparing it to the block mean:

$$X_i' = b \text{ if } X_i \geq \bar{X}, \text{ and} \tag{3-31a}$$

$$X_i' = a \text{ if } X_i < \bar{X}, \text{ and} \tag{3-31b}$$

The values of a and b can be chosen to preserve $\bar{X}$ and σ^2 of the original data by solving two equations:

$$m\bar{X} = (m-q)a + qb \tag{3-32}$$

$$mV^2 = (m-q)a^2 + b^2 \tag{3-33}$$

where q is the number of pels whose value is greater than $\bar{X}$. The values of a and b that satisfy these equations are

$$a = \bar{X} - \sigma\sqrt{\frac{q}{m-q}} \tag{3-34a}$$

$$b = \bar{X} + \sigma\sqrt{\frac{m-q}{q}} \tag{3-34b}$$

Notice that $a \geq \bar{X}$ and $b \leq \bar{X}$, so the reconstructed data jumps up-and-down near the mean in a

fashion that preserves the second moment. If the receiver is to be able to reconstruct the coded block of data, the values of $\overline{X}$ and σ also must be transmitted along with the "ones and zeroes" of the quantizer. Delp and Mitchell used eight bit quantizers to code $\overline{X}$ and σ for blocks of size 4×4. Their example accounts for a total coding rate of 32 (16+8+8) bits/block or 2 bits/pel.

The least rate that can be attained using BTC is 1 bit/pel, the output rate of the one bit quantizer. Since the values of σ and $\overline{X}$ vary from block to block, some overhead must be expected. The amount of overhead depends upon the blocksize and the number of bits used to code σ and $\overline{X}$.

A modified BTC algorithm is used as part of one of the Mixture Block Coding (MBC) systems for monochromatic images in the computer simulations of Chapter 5. The algorithm uses a more sophisticated function to estimate the mean than has been shown above. For BTC the background estimate of the block is a flat surface whose value is set equal to the block mean. The modified algorithm allows the background estimate to be a smooth non-flat surface by using the MBC transform coefficients.

In review, several ideas important to image data compression as they apply to the material of this dissertation have been discussed in this chapter. If image data is to be compressed to a rate that is less than the entropy rate some degradation in the image must be accepted. One way to induce this degradation in an orderly fashion is through quantization. When coding images, the performance of the quantizer may be enhanced by using transformation techniques. Block truncation coding can also be used to advantage in a lossy image source coder. Each of these techniques will be used in the mixture blocks coding and progressive transmission methods developed in Chapter 5. Before moving to the material of Chapter 5, the use of progressive transmission image coding is discussed in Chapter 4.

Chapter 4.

Progressive Transmission of Images

In this chapter the concepts of progressive transmission are introduced and motivated. Some representative techniques found in the literature are discussed. Then, in Chapter 5, the progressive transmission scheme developed for this dissertation will be presented.

Unlike the more conventional coding methods that pass through the source image only once, a progressive coder passes through the image many times. In each pass more detail of the original image is sent to the receiver. The reconstructed image is updated as a whole during each pass. There is no need to wait for the entire image to be coded before a full-field approximation of it can be sent to the receiver. The first-pass image is sparsely coded and can be made available almost immediately. The delay required before this first field can be regenerated at the receiver is only dependent upon the time it takes to code and transmit the first pass. The ability to send a full-field image quickly, even if it is a crude image, is very useful.

It has been shown [61,62,63] that the progressive transmission of image data over low-bandwidth channels is esthetically more appealing than the conventional pel-by-pel or block-by-block transmission methods. In the later cases, the image is coded by scanning each image segment only once. Then the coder moves on to the next segment, never to return again. Usually, the "code and go" methods lead to an image reconstruction that is scanned from side-to-side going from top-to-bottom. When using a low-bandwidth channel data transmission takes a long time, and the image is updated very slowly. In fact, the process can be slow enough to cause the observer's attention to fix itself upon the scanning process instead of staying with the image. The observer is found to be studying coding artifacts instead of the image, and the entire process can become unproductively tedious.

In contrast, when using progressive transmission the source data is coded quickly, and only the gross features of the data are transmitted. The receiver can reconstruct a crude approximation of the image field in a very short time. Thus, there is a complete image, albeit a poor one, available at the receiver ready for viewing at the earliest possible time.

In additional passes, the data treated superficially (or ignored completely) in the first pass, is coded more carefully and sent to the receiver again. The receiver updates its previous version of the image, increasing the amount of visible detail. An update is done on the full image field for each pass so the image seems to improve as a whole, without the undesirable side effects as are found with the single pass coding methods. This iterative process continues until some predetermined stopping criterion is met.

There are stopping criteria that divide progressive transmission source coders into two distinct categories. The first criterion requires that the source image be reconstructed without error. In this case, progressive transmission is used to implement a lossless source coder. The second criterion is not as strict, the transmission of data can be stopped at any time and the reconstructed image does not need to be a perfect replica of the source image. This is a lossy coding scheme and is useful when low data rates are more important than perfect image reconstruction.

Most of the work found in the literature is concerned with lossless progressive transmission. But, these methods can all be terminated early making them useful as lossy coders. Since PT coders pass through the same data repeatedly and transmit information about the same image region many times, they usually require more channel capacity to transmit an image than other coding schemes. Therefore, the positive features of progressive transmission offset its channel inefficiency. Depending on the specific method used the additional overhead can range from a few percent to 50 percent or more. As will be shown later, the PT coder of this dissertation requires little or no overhead when coding at low and medium coding rates.

The next section presents some progressive transmission examples found in the literature. They are selected to show the variety of methods available and are divided into two different coding philosophies, whether the image is coded with fixed or variable blocksizes. They are used to introduce the progressive transmission adaptation for the mixture block coding methods given in Chapter 5.

4.1 Progressive image coding examples from the literature

Four examples of fixed blocksize progressive transmission are discussed below. They are based upon transmitting the lower-frequency data first. Then four variable-blocksize progressive transmission methods are discussed. They are all quasi-filtering methods where the blocksize determines the "frequency" range of transmission. Larger blocks code lower frequency ranges and smaller blocks code greater ones.

4.1a Fixed blocksize methods

Nang [53] blocks the image and computes the transform coefficients for all the blocks and successively passes them to the receiver from lowest to highest frequency order. Thus creating a lowpass version of the image as early as possible. The higher frequency coefficients are transmitted progressively.

Wang and Goldberg [46] use a fixed blocksize transform coding [54,55,56] of the image as a starting place for a lossless progressive coder. In the first pass, the transform coefficients are LBG vector quantized and sent to the receiver. Then the quantized transform coefficients are subtracted from the original coefficients and these coefficients are again vector quantized, with a new code book that is build to match the expected residual coefficient profile of the image. Their subtract-and-code process is continued, coding a new set of residual coefficients at each pass. The residuals at each stage are orthogonal to the previous stages and the residuals become more and more decorrelated. In the final pass, the decorrelated residuals are entropy

coded and sent to the receiver. They suggest that approximately eight passes offers the least overall coding rate. Since new LBG code books need to be built for each pass their method would be expensive to implement.

Dubois and Moncet [57] develop a progressive transmission method for transform coding composite NTSC (color) still images by using a variable-rate transform coding method developed in another of their papers (e.g., [58]). In their method, all transform coefficients that fall below a given threshold are set to zero. The transform coefficients are coded by scanning through them and sending all the non-zeroed coefficients using a Huffman symbol set, and coding the zeroed coefficients using another 0-runlength Huffman symbol set. All coefficients are coded using scalar quantizers. In one experiment they use variable length coding [58], and in another use fixed length coding but varied the quantizer step size [59].

Hoffman and Troxel [65] describes a method where progressive coding is terminated early based upon a want/don't-want user response from a telebrowsing-like system. As mentioned above, many progressive methods require extra bandwidth to complete the transmission of an image. This system relies on the fact that most images are to be browsed are terminated early, as don't want images, and are not completely transmitted, thereby, saving channel bandwidth. When an image is asked for, the first-pass approximation is transmitted as a lowpass representation that is constructed from the output of a subsampled 2-D gaussian filter. The subsampling rate and filter standard deviation are chosen so that 60% of the image energy coded is within each pass. The subsampled filter coefficients are then DPCM coded for the first pass, and PCM coded for additional passes. The filter coefficients are transmitted using an entropy code. The main disadvantage of this method is that the filter characteristics are different for each image and must be designed by trial and error. Since their system is used for telebrowsing this problem was not considered significant because most of the coding process can be done off-line.

4.1b Variable blocksizes methods

Sloan and Tanimoto [47,48] use "pyramid" data structures to sequentially code the image using smaller blocks. Each block is coded to represent some function of the pels it contains (such as minimum value, maximum value, or mean value). Progressive transmission is achieved by using a finer pyramid structure at each pass that codes more detail of the image. Their method uses approximately 33% more data than nonprogressive coding.

Adelsen and Burt [50,52] define a pyramid quasi-filter structure that approximates a pseudo-lowpass filter in the first passes through the image data. They use filters with a higher "cutoff" frequency by using smaller pyramids in the subsequent passes. In each level the current pyramid outputs are subtracted from the previous pass data, thus decorrelating the passes. Kronander [51] offers a method for improving the coding efficiency of the pyramid method. The major disadvantage of their method involves the need to save every pass of the image until it is completely coded; since the image is transmitted in reverse order. Thus, all filtering computations must be complete before the image transmission can begin. This means there is an inherent delay in the time required to receive the first coded symbols.

Knowlton [49] uses a block structure to progressively transmit images. He codes the block means in the first pass, and in subsequent passes divides the blocks into two pieces and codes them using a "differentiator" function. The differentiator value is found using a lookup table that is indexed using the data of the previous pass and the data of the original image.

Dreizen [60] also codes images by partitioning them into a quad tree. Each branch of the tree is subdivided using a busiest-information measurement that is based upon the difference between the maximum and minimum pel inside the considered block. If the measurement is greater than some predetermined threshold, the block is subdivided into four pieces. The blocks are coded uniformly using a representative pel value. It is chosen to be a function of the pel value found in the upper righthand corner of the block. Data compression is enhanced by taking

advantage of the expected correlation between representative values of a subdivided block, and by using Huffman coding for the block representatives. The process terminates if all the coded blocks meet their associated threshold, or if the smallest blocks coded consist of single pels. A modification is also mentioned that uses a BTC method based upon [64].

In review, progressive transmission has been introduced in this chapter through several example systems taken from the literature. Progressive transmission is an image coding technique that gives to the user a sequence of coded versions of the original image that increase in perceptual quality as each new image is received. The coded image from one pass is updated in the next. This allows the user to receive a full-field version of the image much sooner than is possible with the more convenient "single-pass" source coding techniques. In the next three chapters the progressive transmission and the mixture block coding techniques developed for this dissertation are developed. In Chapter 5, these methods are outlined and used to code several example images. Theoretical consideration that are important to the design of such source coding systems are developed in Chapters 6 and 7.

Chapter 5.

Mixture Block Coding and Mixture Block Coding with Progressive Transmission

In this chapter, the mixture block coding systems developed for this dissertation are presented. The MBC systems are used for both single-pass image reconstruction and for multi-pass image reconstruction. The multi-pass systems are modifications of the single-pass systems that allow for progressive transmission. The universal design assumptions that are used for all of these systems and the specific requirements of each different system are discussed. Examples are included to demonstrate their viability as low-rate source coders.

Mixture block coding is a variable-blocksized transform coding technique that codes each image region with different sized blocks. Low-complexity regions are coded with large blocks, and high-complexity regions are coded with small blocks. The number of blocks used to code an image is dependent upon the complexity of the original image, the complexity of the coder used for each blocksize, and the desired quality of the reconstructed image.

The advantage of using mixture block coding is that coding resources are used only as they are needed. The more complex image regions are coded with smaller blocks, so more bits are used to code each pel. Regions of low complexity are coded with large blocks and less bits per pel. This approach is a departure from other coding methods, such as DPCM (Differential Pulse Code Modulation) and standard transform coding [1], which use the same number of bits to code every region of the image without regard for its complexity.

Variable blocksize source coding can be found in the resent literature (e.g., Dreizen [60] or Vaisey and Gersho [66]). The method of this dissertation is similar to the method of Vaisey and Gersho since both methods use a distortion-threshold quad tree structure but, the method developed here addresses progressive transmission while Vaisey and Gersho method does not.

The method of this dissertation also includes a modified block truncation extension, while Vaisey and Gersho is strictly transform coding method.

In Section 5.1, the selection of the largest and smallest blocksizes used to code the image and how the different blocksizes effect the average coding rate is discussed. In Section 5.2, the details of the different MBC and MBC/PT methods and the various quantizers used to code the computer examples are discussed. In Section 5.3 the computer results are given and several important considerations are discussed.

When using MBC methods, the image is first coded using the largest blocksize transform coder. The difference between the original and coded image is measured for each block, and if the residuals fall below a given threshold the block is not coded further. If the block fails the threshold test it is divided into four smaller blocks and recoded. The code-and-test procedure is repeated for each of these smaller blocks. The blocks are recoded until they meet the threshold criterion set for that blocksize or until all blocks that do not meet the threshold criterion are those of the smallest blocksize. The smallest blocks are transmitted as they are, with no further attempt made to improve their coded quality.

The complexity of the original image will determine the number of blocks used to code it. Large blocks will adequately code regions that have little complexity. Large blocks used to code complex regions that fail the threshold test are divided into smaller blocks. Therefore, more blocks are used in an attempt to improve the coded image representation in these more difficult regions. Difficult images are coded with more blocks than less difficult images. Since the number of bits used to code an image is dependent upon the number of blocks used, the more complex images will require more bits than the simpler images. A mixture block coder adapts the number of coding bits used to represent an image to match its complexity.

In general, there are several specific features of the MBC method that must be determined before it can be used to code an image. They include:

- the complexity for the coder used for each blocksize must be chosen,
- the minimum and maximum blocksizes must be selected,
- the block distortion measure must be chosen and
- the block threshold levels must be selected.

The consequences of each of these choices and the selection made for the simulations used to demonstrate the feasibility of MBC are discussed below. Before going into the details of the specifics of the example implementations, the general algorithm for implementing the MBC method will be discussed. Once this is done, how the base algorithm is implemented for simulations will be presented.

The details of MBC are presented through a simple coding example. It is worked through by hand to demonstrate the tree structure of MBC. Then the modifications needed to use progressive transmission with MBC and MBC/PT are introduced. Before the computer results are presented, the details of the simulation MBC method are discussed along with the various quantizing schemes used within these implementations.

5.1 Design considerations

A complex block coder is one that codes a large amount of information about the blocks it codes in a very accurate manor. For transform block coders, a larger number of transform coefficients would be coded for a complex coder, and a small number of coefficients for a simple coder. A low-complexity block coder transmits very few coefficients that can be coarsely quantized. The more coefficients that are coded, and the more bits that are assigned to code them, the less likely it is that the block will fail its distortion threshold and be recoded with smaller blocks. Since smaller blocks probably require more channel capacity on a per pel basis it is desirable to use a complex coder for the larger blocks to save channel resources. But, if the images to be coded contain large areas of small detail, then coding large blocks with too many coefficients wastes channel resources because these image segments are overcoded. It is apparent

that there is a tradeoff in block complexity and the total number of blocks required. The previous chapter discusses some of the mixture block coding methods available in the literature.

5.1a The largest blocksize

The largest blocksize, and the complexity of its coder, determines the lowest coding rate for any image. If an image is of very low detail, for example, a black-and-white image that is monotonically gray, then a single block with one coefficient could be used to code the entire image. Generally, an image has some detail even in its regions of least complexity and very large blocksizes are most likely not of much use since they will almost always be subdivided. The continual subdividing of blocks that are too large causes the coder to waste time that could have been more wisely spent if it had started with a smaller blocksize. Usually a blocksize of 16×16 is about as large as is needed when using more than a single transform coefficient to code the blocks.

The number of bits assigned to code the coefficients of the largest blocksize determine the minimum rate at which an image can be coded. For example, if 16×16 is the largest blocksize used to code a 512×512 image, and a total of 24 bits are used to code each of these blocks, then the entire image can be coded with

$$24\frac{512 \cdot 512}{16 \cdot 16} = 24{,}576 \text{ bits or } 0.0938 \text{ bits/pel} \qquad (5\text{-}1)$$

There is a tradeoff between the lowest expected coding rate and performing unnecessary work recoding blocks that are continually too large.

5.1b The smallest blocksize

The smallest blocksize determines the fine-detail visual quality that is achievable around important image features. Since the coding blocks are square, it is possible to find staircase blockiness in the coded edges of spherical objects (e.g., eyes) if the minimum blocksize is too large. Usually, unless a large number of coefficients are coded per block in the large blocksizes,

the smallest blocksize needed is either 4×4 or 2×2. Some authors use blocksizes as small as 1×1 but are usually coding blocks with only a single value [60].

The smallest blocksize not only determines the quality of local detail that can be obtained, but it determines the maximum coding rate that can be expected when coding very complex images. For example, if 2×2 is the smallest blocksize used and 24 bits are used to code these blocks, then a 512×512 image can be coded with at most

$$24\frac{512 \cdot 512}{2 \cdot 2} = 1{,}572{,}864 \text{ bits or } 6.0 \text{ bits/pel} \qquad (5\text{-}2)$$

This total (and that needed for the minimum coding rate) does not reflect the overhead required to tell the receiver of the actual block structure used to code any particular image. The details of the tree structure used to code the various levels of blocks is discussed next and the overhead required to use such a structure is presented. The idea to be noticed here is the range of coding rates that can be achieved by a MBC system. If the effects of overhead are overlooked an MBC system that uses the largest and smallest blocksize coder indicated above will code an image with a rate ranging from about 0.1 to 6 bits/pel. The exact rate depends upon the complexity of the image being coded and the desired quality required in its final coded form.

5.1c Quad tree structure and the MBC coding rate

The different blocksizes used to MBC or MBC/PT code an image are interconnected using a quad tree structure. The amount of overhead required to code a quad tree and how the overhead effects the average coding rate is discussed. There is no overriding reason for using a quad tree structure but it was found to be useful in the computer simulations.

The quad tree (see for example, Figure 5.1) algorithm used here divides a given block into four smaller blocks when the distortion threshold is exceeded. The operation of the tree structure is presented by stepping through the details of an example. Consider the coding of a 16×16 block with a minimum blocksize of 2×2.

The 16×16 block is coded and the distortion level of the block is measured. If the distortion is greater than the predetermined maximum level for 16×16 blocks, dmin(16×16), the block is divided into four 8×8 blocks for additional coding. After each of the 8×8 blocks is coded their resulting distortion levels are compared with the 8×8 distortion level, dmin(8×8). If any fail to meet the distortion threshold, they are divided into four 4×4 blocks for additional coding. The process is continued until the only image blocks not meeting their given distortion threshold are those of size 2×2. Since 2×2 is the smallest allowed blocksize, these blocks are coded and transmitted de facto, making no further attempt to improve their distortion level.

Consider the example coding of a 16×16 image block shown in Figure 5.2a. For clarity, let the four sub-blocks of an arbitrary block be numbered as shown in Figure 5.2b. Let the block distortion thresholds be:

$$\text{dmin}(16\times16) = 12,$$

$$\text{dmin}(8\times8) = 12 \text{ and}$$

$$\text{dmin}(4\times4) = 10.$$

After coding, the 16×16 block it is found that, say, d(16×16)=30, so it must be divided and recoded as four 8×8 blocks. Here, as shown in Figure 5.2, the four coded 8×8 blocks have distortion levels of 10, 15, 4, and 6. Now, since one of these 8×8 blocks fail to meet dmin(8×8), it must be divided and recoded as four 4×4 blocks. Since the 4×4 distortions levels for the divided 8×8 block are 10, 8, 15, and 6; only one of these 4×4 blocks fails to meet dmin(4×4). The block is recoded as four 2×2 blocks.

The above coding process is depicted as a quad tree structure in Figure 5.1. There is

one 16×16 block,

four 8×8 blocks,

four 4×4 blocks and

four 2×2 blocks,

for a total of 13 blocks of coded information which are connected together through the use of a relative pointer scheme. To guarantee the receiver reconstructs the image correctly, a bit map of side information is sent along with the block coefficient information. One bit of side information is needed for each block. If a block is to be divided, its bit value is set to 1; if not, its bit value is set to 0. To tell the receiver the ordering of these 13 coded blocks, 9 bits of side information are needed: 1,0100,0010. (The first bit shows the 16×16 block is divided into 8×8 blocks. The next four bits indicate only the second 8×8 block is divided. The last four bits indicate the state of the 4×4 blocks are obtained from the divided 8×8 block. They show only the third 4×4 block is divided into 2×2 blocks. Notice the 2×2 blocks of this example are placed with the bit maps generated at the 4×4 level, so no side information is needed to code them.

It can be seen that there is one overhead bit per block in each pass, except for the blocks of the last pass which require no overhead because they cannot be divided. Consider the case where an MBC system that has more than one pass is used.[1] Let the number of blocks tested in the i-th pass be b_i, and let the number of blocks that fail the threshold criterion be f_i and the let the number that do not fail be p_i. Notice it is true that $b_i=f_i+p_i$. The number of bits needed to code the first pass is

$$p_1 r_1 + b_1 \tag{5-3}$$

where there are b_1 bits of overhead to tell the receiver which of the first pass blocks have failed and each block is coded with r_1 bits. If a quad tree structure is used to divide the blocks that fail the first pass, then the number of blocks in the second pass is $b_2=4f_1=f_2+p_2$. The number of bits needed to code the second pass are

$$p_2 r_2 + b_2 = p_2 r_2 + 4f_1 \tag{5-4}$$

---

[1]If the coder has only one pass then number of overhead bits is zero. In fact, such a system cannot be considered a mixture block method at all, except as a degenerate case.

The number of bits needed to code each subsequent pass is

$$p_i r_i + b_i = p_i r_i + 4f_{i-1} \tag{5-5}$$

except for the last pass (n-th) which has no overhead. If we define b_n to be 0, the total number of bits to code the image is

$$B_{MBC} = \sum_{i=1}^{n} p_i r_i + b_i \text{ bits} \tag{5-6}$$

and the average coding rate is

$$R_{MBC} = \frac{B_{MBC}}{N} \text{ bits/pel} \tag{5-7}$$

where N is the number of pels in the image. The total number of overhead bits, b_o, is

$$b_o = f_1 \qquad n=2 \tag{5-8}$$

$$b_o = f_1 + \sum_{i=2}^{n} 4f_{i-1} \qquad n>2 \tag{5-9}$$

Consider the minimum rate at which an image can be coded using MBC, it is obtained when the entire image is coded with a single pass and occurs when the distortion threshold is set high enough to guarantee that all of the image is coded with the largest blocksize. When this is true, (6) reduces to (3) and the minimum rate for (7) is

$$\min R_{MBC} = \frac{b_1 r_1 + b_1}{N} \tag{5-10}$$

If it were known ahead of time that the image was to be coded in a single pass the overhead bits of (10) would not be necessary. But, for the general multipass coder this is not the case. For a multipass coder the overhead bits are all set to 0 and the receiver would know that no smaller blocks are to be transmitted. Using the example of section 5.1a, the minimum rate when coding a 512×512 image is obtained by coding the image with 1024 16×16 blocks

$$\min R_{MBC} = \frac{1024 \cdot 24 + 1024}{512^2} = 0.0977 \text{ bits/pel} \tag{5-11}$$

The maximum rate that can be used to code an image using MBC is obtained when the threshold is set to guarantee that all blocks are divided into the smallest allowed blocksize. (This can be guaranteed if d_{min} is set to 0 for all blocksizes.) All of the overhead bits for all passes are set to 1, and (7) becomes

$$\max R_{MBC} = \frac{b_n r_n + \sum_{i=1}^{n} b_i'}{N} \tag{5-12}$$

where the b_i have been set to their maximum values, b_i'. The b_i are maximal when the entire image is coded with the i-th blocksize. If the i-th blocksize is n_i^2 then

$$b_i' \equiv \max b_i = N/n_i^2 \tag{5-13}$$

Using 16×16 as the largest blocksize, the 2×2 smallest blocksize example of section 5.1b gives a maximum coding rate of

$$\max R_{MBC} = \frac{1{,}572{,}864+(1024+4{,}096+16{,}384)}{512^2} = 6.0820 \text{ bits/pel} \tag{5-14}$$

If the original image can be divided into N_i blocks in the i-th pass, the coded image percentage for this pass is

$$P_i = \frac{p_i}{N_i} \tag{5-15}$$

Since the pels of the image are coded in only one of these n passes it must be true that

$$\sum_{i=1}^{n} P_i = 1 \tag{5-16}$$

In contrast to (16), notice that a progressive system codes complex regions of the image with more than one pass. This fact shows that (16) is not true for MBC/PT. The coding rate and overhead for MBC/PT will be discussed in the next section.

5.1d The MBC/PT coder and its coding rate

Notice that the tree structure of the last section made no mention as to whether each 16×16 block was coded as a whole from the largest to smallest blocksize, or whether the entire image is coded with the larger blocksize before coding any of these blocks that fail the threshold test with the next smaller blocksize. For MBC it does not matter which of these methods is used. But, the latter method is particularly useful when designing a progressive transmission coder.

If the entire image is passed through for each blocksize and the difference image is saved for additional coding in the next pass using a smaller blocksize, the MBC method can be used as a progressive transmission coder. If each pass is transmitted by the receiver as soon as it is

available, the receiver can reconstruct a crude representation of the original image using the larger blocksizes while the MBC/PT coder is processing the smaller blocks of the next pass.

The information received from coding the difference image in each subsequent pass is reconstructed using smaller blocks. The received image is updated with these smaller blocks and it acquires more clarity with each pass. There are no serious problems in updating MBC for use as a progressive transmission coder. If each MBC pass is successively transmitted then progressive transmission is immediately available. Since the first pass is coded with very few bits the receiver has an image, although a crude image, almost immediately. If the successive passes are caused to code the difference image, instead of the original, the reconstructed image can be updated differentially at each pass. Thus, with this modification progressive transmission can be achieved.

As is the case for MBC, MBC/PT updates only those image regions that have coded poorly in previous passes. Only those regions of the image which need additional coding continue to use coder resources. The regions of the image with low detail are coded quickly, and remain fixed, while the rest of the image continues to change as the information for each pass is received.

The high-detail regions of an image are coded more than once when using the MBC/PT method, so they require a greater channel capacity to transmit their coefficients. MBC/PT uses more channel resources to code high-detail regions because it must transmit data that codes the final pass and every previous pass. This means that if the MBC/PT coder uses the same number of blocks and the same number of bits to code each block as an MBC coder, the image will be coded with more total bits. This is offset by the fact that the original image can be converged to more quickly when using MBC/PT because the busy regions are coded with more than a single pass of information. Therefore, it is possible for MBC/PT to use fewer blocks to code an image than MBC. Therefore, it is not obvious which system will code an image with

fewer bits.

Let the same definition for r_i, f_i and p_i as was used for MBC apply here. Then number of bits needed to code the first pass is

$$b_1 r_1 + b_1 \tag{5-17}$$

Notice that all blocks are coded in the first pass, therefore $b_1 = b_1'$. Every MBC/PT block that is failed in the previous pass is recoded as a subdivided block in the new pass. Using the quad tree structure, the number of blocks in the i-th pass ($i>1$) is $b_i = 4f_{i-1}$, and the number of coding bits are

$$b_i r_i + b_i = 4(f_{i-1} r_i + f_{i-1}) \tag{5-18}$$

The total number of bits needed to code the entire image found by summing over all coding passes,

$$B_{MBC/PT} = \sum_{i=1}^{n} b_i r_i + b_i = (b_1 r_1 + b_1) + \sum_{i=2}^{n} 4(f_{i-1} r_i + f_{i-1}) \text{ bits} \tag{5-19}$$

where the sum is zero when n=1. The final coding rate is

$$R_{MBC/PT} = \frac{B_{MBC/PT}}{N} \text{ bits/pel} \tag{5-20}$$

where N is the number of image pels.

The total number of overhead bits for an MBC/PT system are

$$b_O = b_1' + \sum_{i=2}^{n} 4f_{i-1} \qquad n \geq 2 \tag{5-20a}$$

The image percentage coded in the i-th pass is

$$P_i = \sum_{i=1}^{n} \frac{p_i}{N_i} \tag{5-21}$$

and since every block that is coded in the i-th pass has been coded in every previous pass it is true that

$$p_i \geq p_j \text{ for } i<j \tag{5-22a}$$

and

$$\sum_{i=1}^{n} p_i \geq 1 \tag{5-22b}$$

where equality occurs only when the image is coded in a single pass.

The minimum coding rate that can be obtained for MBC/PT when only the first pass is coded. In this case, (20) becomes

$$\min R_{MBC/PT} = \frac{b_1 r_1 + b_1}{N} \tag{5-23}$$

which is the same as the rate that was obtained for MBC (10). For similar block coding types, it is true that the minimum MBC and MBC/PT rates are always equal

$$\min R_{MBC} = \min R_{MBC/PT} \tag{5-24}$$

Using the example of section 5.1a, the minimum rate can be computed to be the same as is found by (11)

$$\min R_{MBC/PT} = \min R_{MBC} = 0.0977 \text{ bits/pel} \tag{5-25}$$

The MBC/PT maximum rate is obtained when all passes are coded without any blocks passing the threshold test. Therefore, the maximum coding rate is

$$\max R_{MBC/PT} = \frac{(b_1' r_1 + b_1) + \sum_{i=1}^{n} 4(b_{i-1}' r_i + b_{i-1}')}{N} \tag{5-26}$$

where the b_i' are the maximal values of b_i as are found by (13). The maximum sum is computed recursively using the maximal b_i for each pass. Noticing for MBC/PT that $b_1'=b_1$, (26) becomes

$$\max R_{MBC/PT} = \frac{\sum_{i=1}^{n} 4^{i-1}(b_1 r_i + b_1)}{N} \tag{5-27}$$

Comparing with (12) shows that when using similar block coder types

$$\max R_{MBC/PT} \geq \max R_{MBC} \tag{5-28}$$

Equality is obtained only when the image is coded with only one pass. (This assumes we avoid the trivial case were $r_i=0\ \forall i>1$.) Using the same block coder types as was used for the maximum MBC rate example, the maximum MBC/PT rate is

$$\max R_{MBC/PT} = \frac{\sum_{i=1}^{4} 4^{i-1}(1024 \cdot 24 + 1024)}{512^2} = 8.3008 \text{ bits/pel} \tag{5-29}$$

Comparing the coding rates of this and the last section may lead one to believe MBC out performs MBC/PT. In general, this is not true since MBC codes any image pel only once,

while MBC/PT can code any pel more than once. The original image may be converged to more quickly with MBC/PT than with MBC. As a result, MBC/PT may code the entire image with fewer blocks, and fewer bits, than can be achieved by MBC.

5.2 The MBC and MBC/PT simulators

The main advantage of MBC systems is that if a large block does not adequately code a given image region, it is divided into smaller blocks and recoded. There is no strict advantage in using a large number of coefficients to code any particular blocksize. In fact, there is a tradeoff between expending more effort coding the larger blocks so fewer smaller blocks are used, and coding the larger blocks minimally to let the threshold algorithm assign more smaller blocks for coding.

For the simulation examples of this dissertation it was chosen to code each block with one of two different methods. The first method uses only four DCT transform coefficients, including the dc and three lowest order frequency coefficients (Figure 5.3). It is used for MBC and MBC/PT of both monochrome and color images. The second method uses these four DCT coefficients and uses a modified form of block truncation coding. The use of BTC limits the lowest rate that can be used to code an image to one bit, so it is not useful at lower rates, but it does offer the ability to easily code sharp edges in the image without using an excessive number of small blocks. Each of these methods is discussed below.

5.2a The DCT block coders

As was mentioned above, it is not necessary to code a large number of transform coefficients when using MBC. The example MBC (and MBC/PT) systems used here were all designed to use only the four lowest order DCT transform coefficients of the coded blocks. This is done no matter what size block is coded; not so much to attain the best overall coding rate, but to strike a median between PT coders which code a minimum of information about a given

block [49] and those which code a large amount of information per block [57]. This accomplishes two things.

In the first place, it shows an image can be adequately coded in a relatively small number of passes (at most four for the examples here) using a small number of transform coefficients at each pass, and secondly, it shows that the coding can be done using a simple algorithm for each pass. In addition, when using the same transform coder for each pass it is also possible to share the same vector quantizer between all of the passes. Using this method saves quantizer design effort. The quantizer design process will be discussed in the next section.

When using a small number of coefficients to code large blocks of data ($\geq 16\times16$) it was found that coding with so few coefficients induced large coding errors at pels near the block edges. Therefore, it is necessary to limit the largest blocksize accordingly. For the computer simulations the maximum blocksize was set at 16×16. Since four coefficients are coded per block this limits the smallest blocksize to 2×2.

Using 2×2 as the smallest blocksize can increase the average coding rate significantly without improving the SNR. But, it was found that the overall visual crispness of the image was improved by using 2×2 blocks. In the final analysis, the choice of the smallest blocksize depends upon the quality desired and the allowed capacity of the system being designed.

5.2b Using block truncation coding with MBC

Block truncation coding can be used to include information about the high frequency content of an image block. The one bit quantizer output can be useful in marking image edges and fast contours in a way that cannot be attained using only low frequency transform coefficients. The BTC method of Delp and Mitchell is modified to allow it to be used with the four coefficient DCT coders used with the MBC systems of this chapter.

Instead of using the block mean as the threshold for the quantizer, an estimate constructed from the four lowest frequency DCT coefficients is used. In this case, the second moment is redefined

$$\sigma^2 = \frac{1}{n^2-1} \sum_{i,j=0}^{n-1} (X_{ij}-X'_{ij})^2 \tag{5-30}$$

where, using the notation of (3-28), the X_{ij} and $\bar{X}_{ij}$ are the original and four-coefficient DCT pel representations in an n×n block of data. The new second moment measures the mean difference between the original block data and its estimate, instead of the block energy in the algorithm developed by Delp and Mitchell. Using the new second moment in a modified version of the Delp and Mitchell preserving equations

$$m\bar{X}_{ij} = (m-q)a + qb \tag{5-31a}$$

$$m\sigma^2 = (m-q)a^2 + b^2 \tag{5-31b}$$

the output of the one bit quantizer for each pel becomes dependent upon its DCT pel estimate, $\bar{X}_{ij}$,

$$X'_{ij} = \bar{X}_{ij} - \sigma\sqrt{\frac{q}{m-q}} \quad \text{if } X_{ij} \geq X'_{ij} \tag{5-32a}$$

$$X'_{ij} = \bar{X}_{ij} + \sigma\sqrt{\frac{m-q}{q}} \quad \text{if } X_{ij} < X'_{ij} \tag{5-32b}$$

The reconstructed data of (32) follows the contour of the DCT estimate instead of only being able to "float around the mean." The modified method includes more information about the original image data than is available when using the Delp and Mitchell algorithm. It codes low frequency information by using the low frequency DCT coefficients, and high frequency information is coded using σ and the output of the one bit quantizer.

But, the increase in block coding quality is not free. Its price comes with the cost of coding the four DCT coefficients, instead of coding the mean alone. The modified BTC algorithm is used in an MBC system later in the chapter.

5.2c The distortion measures

Any distortion measure can be used to drive a MBC system. The requirements of some of the different distortions measures are discussed in Chapter 2. It is possible to use different distortion measures for each different blocksize of an MBC system to adjust for the expected radial frequency coding sensitivity of the eye. Each different blocksize represents a different spatial frequency range that is to be coded, and details of distortion induced within each of these blocksizes will be seen differently by the eye. The details of such an implementation were outside of the hardware capabilities of the facilities used to test the quality of the final coded images. Therefore, simple distortion measures were used to build the MBC system of this dissertation.

Two different distortion measures were used in the MBC systems studied here. They depend upon whether the images coded were monochrome or color. The monochrome distortion measure used for the examples is the maximum absolute difference:

$$d_{bw} = \max_i |x_i - y_i| \tag{5-33}$$

where the range of i is taken over the entire block be coded, and y_i is the coded value of the original image pel x_i. For color images maximum mean square difference is used

$$d_c = \max_i \sqrt{(x_i - y_i)^T (x_i - y_i)} \tag{5-34}$$

where y_i is the coded value of color pel x_i.

The maximum absolute difference was found to be useful when coding monochrome images because it simultaneously allowed the coder the offer a low coding rate and a higher mse. When this distortion measure was used to code color images all three colors have to be included,

$$d_{c,abs} = \max_i (|x_{i1} - y_{i1}|, |x_{i2} - y_{i2}|, |x_{i3} - y_{i3}|) \tag{5-35}$$

where the 1, 2 and 3 subscripts indicated the first, second and third color component of the color format being used (RGB or YIQ) it was found that a single color component could dominate the distortion measured for a block. This could force the coding rate to be excessively high because the measure would cause too many blocks to be subdivided unnecessarily. The mean square

difference distortion measure alleviated the problem. Since the Y-component of the YIQ format dominates the other two components it was felt that a component weighting scheme

$$d_q = \max_i \sqrt{(x_i - y_i)^T W (x_i - y_i)} \tag{5-36}$$

where W is a 3×3 diagonal weighting matrix of the form

$$w_{ii} > 0 \qquad i = 1,2,3$$

$$w_{ik} = 0 \qquad i,k = 1,2,3 \text{ and } i \neq k$$

might improve the performance of the coder, but the method was not adequately tested to verify this conjecture.

5.2d The quantizers

Two general types of quantizers are used in the MBC and MBC/PT simulations of this chapter: scalar quantizers (SQs) and LBG vector quantizers (VQs).

For the computer simulations, scalar quantizers are used in two ways. Firstly, they act as a benchmark for comparison with their vector quantized counterparts. A VQ should code with less rate than a SQ given the same distortion level. Secondly, since LBG quantizers are designed using a training set, the scalar quantized coders are used as an integral part of the source coder from which the VQ training set is taken. The quantizers used will be discussed in three sections each relating to the color type of data to be quantized (monochrome, RGB and YIQ).

The following presentation is started with a discussion of the quantizers used to code the monochrome image transform coefficients. Then the quantizer used to code RGB coefficients are discussed and, finally, the YIQ color coefficient quantizers. Since there are many different categories within each color type, each is divided into three subcategories for the convenience of presentation:

- SQ MBC scalar quantized MBC,
- VQ MBC vector quantized MBC and
- SQ and VQ MBC/PT SQ and VQ MBC with progressive transmission.

The last category combines the scalar and vector quantizer used to code MBC/PT because these systems are straightforward extensions of the two MBC categories. An important idea that should be kept in mind when studying these quantizers is that each source set must be matched to the quantizer used to code it. The following diagram indicates how this is done.

$$Y \rightarrow \boxed{\text{Scale: S}} \rightarrow \boxed{\begin{matrix}\text{Unit Variance} \\ \text{Quantizer: Q}\end{matrix}} \rightarrow \boxed{\text{Rescale}} \rightarrow \bar{Y}$$

$$\bar{Y} = S^{-1}Q(SY)$$

The scale factor, S, is used to match the variance of the source set, {Y}, to that of the quantizer. The scale factor is needed to insure maximum quantizer performance. Every different source coder will, in general, have a different set of scale factors.

Before delving into the details of the particular functions of each of these quantization systems, an introduction to the three color types is made. The following material is meant to complement the presentation of Chapter 3. The goal is to point out the differences and similarities that are to be found between the different color types and show how these facts are important to the design of the quantizers used in this chapter.

Monochromatic data is usually represented with non-negative data values where the pel value is proportional to the source light intensity. All of the monochrome images used here are represented with 8 bits of non-negative intensity ranging from 0 to 255. Since it is hard to predict the source statistics of the dc coefficient of image blocks no particular source dependencies are built into the quantizers that code them. To compensate for this fact, the quantizers used to code the dc coefficient for the first time will be an 8-bit uniform scalar quantizers (8-USQs). After the dc coefficients have been coded once, their MBC/PT residuals are easy to predict and more specialized, and lower rate, quantizers are used to code them. Typically, a 5-bit optimal laplacian scalar quantizer (5-OLSQ) is used. It was found that if a 5-bit optimal gamma scalar quantizer was used instead of the 5-OLSQ that similar results were obtained.

Henceforth, all such quantizers will be called 5-OLSQ. In that which follows the abbreviation BW (black and white) will be used to indicate that monochromatic images are being coded.

The three RGB colors are also non-negative intensity values that represent the light intensity that falls within certain passbands of the human visual system. The three colors can be mixed together to form a monochromatic image, but to get a good monochromatic representation all three colors must be included. All three components represent a color band as it is seen by the cones of the eye. If any one color is missing, the monochrome representation has the appearance of being band-rejection filtered. Of course, the missing color band is the missing R, G or B color component. The important feature of RGB coding is that all of the color components are non-negative and have similar expected coefficient mean and variance values. This causes the RGB quantizer systems to be symmetric with respect to the three colors. This is not true for the YIQ coders. The RGB color components have nearly the same expected intensity and variance values, and for the image used here are coded with 8 bits of resolution (0-255). At least for the their first coding, an 8-USQ is used.

The YIQ color components, as was mentioned in Chapter 3, are divided into the luminance (Y) and chrominance (I and Q) representations of the original image. The YIQ color system luminance component has the same properties as are found with monochrome images, and the chrominance components have less variance, typically 1/3 to 1/2 of that of the luminance, and are typically zero mean. This indicates that luminance may need to be quantized differently than the chrominance, and the three color symmetry of the RGB color system is lost. In that which follows the differences between luminance and chrominance will effect the quantizer designs. Typically, at least for the MBC systems, the luminance dc coefficients will be coded with 8-USQs and the chrominance dc coefficients with 5-OLSQs.

Monochrome image quantizers

BW SQ MBC

As has been already mentioned, all the blocksizes in the computer simulations are DCT coded with four coefficients, including the dc and three lowest frequency (ac) coefficients. When MBC was used with monochrome images, the dc coefficients were coded with an 8-bit uniform scalar quantizer that codes the levels from 0 to 255. The ac coefficients were mapped into a 5-bit unit variance optimal laplacian scalar quantizer. Since the dc coefficient quantizer matches the default bit allocation of the source (8 bits with levels 0-255) they do not need to be scaled. The variance of the ac coefficients varies with blocksize and each are scaled appropriately.

Let the coefficients of the i-th MBC pass be $Y_{kl}^{(i)}$ where the double subscripts indicate the DCT coefficient index as is shown by (3-28). The dc coefficient is $Y_{00}^{(i)}$, and the ac coefficients are $Y_{01}^{(i)}$, $Y_{10}^{(i)}$ and $Y_{11}^{(i)}$. The four quantizers of each coder pass can be represented by

$$\bar{Y}_{00}^{(i)} = Q_{dc}(Y_{00}^{(i)}) \tag{5-37}$$

$$\bar{Y}_{kl}^{(i)} = S_i^{-1} Q_{ac}(S_i Y_{kl}^{(i)}) \quad kl = 01,\ 10 \text{ and } 11 \tag{5-38}$$

where the S_i are the ac quantizer scaling factors for the i-th pass. The *dc* subscript indicates the quantizer is an 8-USQ, and the *ac* indicates a 5-OLSQ. Even though each of the three ac coefficients of the same coding pass have different variances, they were found to be of close enough value that all could be coded with the same scaling factor. This was found to be true for all of the coders studied in this chapter, and it is used implicitly in the sequel. Table 5.1 show the values of the S_i used for the BW MBC image simulations.

BW VQ MBC

The LBG vector quantizers used with BW MBC were designed using a training set that was taken from the unquantized transform coefficients of scalar quantized BW MBC images. Three images, that were different from the image coded in the examples below, were coded to

supply the training set. The process used to construct all of the LBG VQs of this chapter is shown in Figure 5.4. Notice that this process is independent of the coder being used, and the dimension size of the code book being designed. Because of these properties, the same algorithm can be used to design all of the vector quantizers of this chapter. For each design the things that need to be specified are the training set and the size of code book to be built. Everything else remains unchanged. The vector quantizers are designed using the split-and-optimize algorithm mentioned in section 3.1b.

The distortion thresholds used to acquire the SQ MBC training set were adjusted to levels lower (tighter) than are necessary to code images for typical picture quality. Typical picture quality desired for the computer simulations is defined by a de facto standard of 30 dB PSNR. This PSNR level is commonly used in the literature for comparing image coding techniques and usually gives an image that has minimal visual quality. A PSNR that is less than 30 usually indicates that the image is coded with clearly visible distortion. The tighter thresholds were set to code the three test images with a typical PSNR value greater than about 34 dB PSNR. Using these levels, approximately 20,000 training vectors were obtained. These rules of thumb were used to design of all the LBG VQ discussed in this chapter. Every vector quantizer code book was constructed using training sets that contained approximately 20,000 training vectors.

The dc coefficient of the VQ MBC was coded separately from the ac coefficients. This is because the dc coefficient tends to act independently of the ac coefficients, and it was felt that tying the dc coefficient to the ac coefficients would not make good use of the vector quantizer. The dc coefficients were coded using the same scalar quantizer (8-USQ) as mentioned above, and the three ac coefficients were coded as a single vector. The VQ code book size was set at 256.

Therefore, the VQ MBC system uses two quantizers to code each pass. By defining the 3-dimensional vector whose components are the i-th pass ac coefficient

$$Y^{(i)} = (Y_{01}^{(i)}, Y_{10}^{(i)}, Y_{11}^{(i)}) \tag{5-39}$$

the quantizers used for the VQ MBC system are

$$\bar{Y}_{00}^{(i)} = Q_{dc}(Y_{00}^{(i)}) \tag{5-40}$$

$$\bar{Y}^{(i)} = S_i^{-1} Q_{3lbg}(S_i Y^{(i)}) \tag{5-41}$$

where the $3lbg$ subscript indicates a 3-dimensional LBG VQ. The scaling factors used here must map the coefficient vectors into the LBG code book that was designed using the scalar quantizer scaling factors of Table 5.1. The same scaling factors that mapped the coefficients of the SQ MBC system into unit variance scalar quantizers will correctly map the coefficient vectors into the LBG VQ code book. This fact is used in the design of the remaining LBG VQs of this chapter.

SQ and VQ BW MBC/PT

The BW MBC/PT quantizers are not much different from the quantizers used for MBC. The only changes come from the fact that MBC/PT codes difference images after the first pass. This means that the dc coefficients of all subsequent passes will be distributed near zero. It was found (by generating histograms) that the subsequent pass dc coefficients were approximately laplacian. Their variance was less than the variance of the associated ac coefficients and to overcome this problem an additional scaling factor was applied to the subsequent pass dc coefficient quantizer.

When using SQ MBC/PT, the dc coefficient quantizer for the first pass was made to be identical to that of (32). The subsequent pass dc coefficients were coded with a 5-OLSQ

$$\bar{Y}_{00}^{(i)} = C_i^{-1} Q_{ac}(C_i Y_{00}^{(i)}) \qquad i>1 \tag{5-42}$$

As a whole, the SQ MBC/PT quantizers are

$$\bar{Y}_{00}^{(1)} = Q_{dc}(Y_{00}^{(1)}) \tag{5-43}$$

$$\bar{Y}_{00}^{(i)} = C_i^{-1} Q_{ac}(C_i Y_{00}^{(i)}) \qquad i>1 \tag{5-44}$$

$$\bar{Y}_{kl}^{(i)} = S_i^{-1} Q_{ac}(S_i Y_{kl}^{(i)}) \qquad kl = 01, 10 \text{ and } 11 \tag{5-45}$$

where the dc coefficient scaling factor are $C_i=4S_i$. The ac coefficients of subsequent passes were not found to be of different enough character from their MBC counterparts to warrant any special allowances. The scaling factors used for the ac coefficients of the BW MBC/PT system were set to the same values as were used for BW MBC.

With the modification of (33) the LBG VQ code book for BW MBC/PT was designed in a similar fashion to the LBG VQ code book for BW MBC. Using the same notation as above, the BW MBC/PT quantizer are

$$\bar{Y}_{00}^{(1)} = Q_{dc}(Y_{00}^{(1)}) \tag{5-46}$$

$$\bar{Y}_{00}^{(i)} = C_i^{-1}Q_{ac}(C_i Y_{00}^{(i)}) \qquad i>1 \tag{5-47}$$

$$\bar{Y}^{(i)} = S_i^{-1}Q_{3lbg}(S_i Y^{(i)}) \qquad \forall i \tag{5-48}$$

where the dc coefficient scaling factors, C_i, are the same as are used with (42), and the ac coefficient scaling factors, S_i, are the same shown in Table 5.1. The LBG VQ code book used for (48) has 64 vectors.

Color image quantizers

When coding color images the four DCT coefficients become twelve, four for each of the three color planes. The computer simulations presented at the end of this chapter code color images using both the RGB and YIQ schemes. As is the case for monochrome images, LBG vector quantizers were used in these simulations. The training sets used for the design of VQ code books were constructed from unquantized coefficients taken from SQ MBC and SQ MBC/PT coded training images. Since the RGB and YIQ quantization systems are different in several ways, each is discussed separately. The RGB systems are discussed first.

RGB color quantizers

RGB SQ MBC

The RGB SQ MBC system codes each of the three color components the same. The dc and ac coefficients of each color plane were coded with 8-USQ and 5-OLSQ, respectively. Therefore, the quantizers of (23) and (24) represent the four SQs used for each color plane, and the full system has a total of twelve SQs. This system is represented by

$$\bar{Y}_{00}^{(i,j)} = Q_{dc}(Y_{00}^{(i,j)}) \tag{5-49}$$

$$\bar{Y}_{kl}^{(i,j)} = S_i^{-1} Q_{ac}(S_i Y_{kl}^{(i,j)}) \qquad kl = 01,\ 10 \text{ and } 11 \tag{5-50}$$

where i is the pass number, $j\varepsilon\{R,G,B\}$ indicates the color component of interest, and the S_i are the ac scaling factors shown in Table 5.2.

RGB VQ MBC

The RGB VQ MBC system was designed to code each of the three dc coefficients using an 8-USQ for each coefficient, and the nine ac coefficients were coded as a single 9-dimensional vector. The 9-dimensional LBG VQ code book was designed, as is discussed above, using the tighter SQ coder distortion thresholds. Since the final LBG code vectors are 9-dimensional (instead of 3-dimensional as was the case for monochrome images) a larger LBG code book containing 1024 vectors was constructed. If the nine color components of a block are grouped to define the 9-vector

$$W^{(i)} = (Y_{01}^{(i,R)}, Y_{10}^{(i,R)}, Y_{11}^{(i,R)}, \ldots, Y_{01}^{(i,B)}, Y_{10}^{(i,B)}, Y_{11}^{(i,B)}) \tag{5-51}$$

the four quantizers of the RGB VQ MBC system are

$$\bar{Y}_{00}^{(i,j)} = Q_{dc}(Y_{00}^{(i,j)}) \qquad j\varepsilon\{R,G,B\} \tag{5-52}$$

$$\bar{W}^{(i)} = S_i^{-1} Q_{9lbg}(S_i W^{(i)}) \tag{5-53}$$

where the ac coefficient scaling factors are shown in Table 5.2.

RGB SQ and VQ MBC/PT

The RGB MBC/PT systems must take into consideration the same difference image characteristics that were discussed for the BW MBC/PT coders. In this case, the difference images consist of three individual color planes. When using RGB SQ MBC/PT, the RGB color symmetry property causes the coder to use three identical SQ MBC/PT coders, one for each color. They are of a similar form to those found in (43), (44) and (45)

$$\bar{Y}_{00}^{(1,j)} = Q_{dc}(Y_{00}^{(1,j)}) \tag{5-54}$$

$$\bar{Y}_{00}^{(i,j)} = C_i^{-1}Q_{ac}(C_i Y_{00}^{(i,j)}) \qquad i>1 \tag{5-55}$$

$$\bar{Y}_{kl}^{(i,j)} = S_i^{-1}Q_{ac}(S_i Y_{kl}^{(i,j)}) \qquad kl = 01,\ 10 \text{ and } 11 \tag{5-56}$$

where $j \in \{R,G,B\}$, the dc coefficient scaling factors are $C_i=2S_i$, and the ac scaling factors, S_i, are found in Table 5.2.

The RGB VQ MBC/PT system grouped the ac-color coefficient into a 9-dimensional vector similar to the one used for RGB VQ MBC. The LBG VQ code book of size 1024. The first pass dc-color coefficients were coded as in (40) using 8-OLSQs, but in the subsequent passes the dc-color coefficients were quantized as a 3-dimensional vector using an LBG code book of size 64. Letting this 3-dimensional vector be defined as a grouping of the dc coefficients

$$Y^{(i)} = (Y_{00}^{(i,R)}, Y_{00}^{(i,G)}, Y_{00}^{(i,B)}) \tag{5-57}$$

the twelve quantizers of the first pass are

$$\bar{Y}_{00}^{(1,j)} = Q_{dc}(Y_{00}^{(1,j)}) \tag{5-58}$$

$$\bar{Y}_{kl}^{(i,j)} = C_i^{-1}Q_{ac}(C_i Y_{kl}^{(i,j)}) \qquad kl = 01,\ 10 \text{ and } 11 \tag{5-59}$$

where $j\epsilon\{R,G,B\}$, and the subsequent passes ($i>1$) are coded using two LBG quantizers

$$\bar{Y}^{(i)} = C_i^{-1}Q_{3lbg}(C_i Y^{(i)}) \tag{5-60}$$

$$\bar{W}^{(i)} = S_i^{-1}Q_{9lbg}(S_i W^{(i)}) \tag{5-61}$$

All of the dc scaling factors are $C_i=2S_i$, and the ac scaling factors, S_i, are shown in Table 5.2.

YIQ color quantizers

YIQ SQ MBC

The scalar quantized YIQ MBC system codes the three color components of each blocksize with twelve quantizers. One 8-USQ was used to code the luminance dc coefficient and all of the remaining quantizers are 5-OLSQs using differing scaling factors:

$$\bar{Y}_{00}^{(i,Y)} = Q_{dc}(Y_{00}^{(i,Y)}) \tag{5-62}$$

$$\bar{Y}_{00}^{(i,j)} = C_i^{-1} Q_{ac}(C_i Y_{00}^{(i,j)}) \quad j\varepsilon\{I,Q\} \tag{5-63}$$

$$\bar{Y}_{kl}^{(i,j)} = S_i^{-1} Q_{ac}(S_i Y_{kl}^{(i,j)}) \quad kl = 01,\ 10 \text{ and } 11,\ j\varepsilon\{Y,I,Q\} \tag{5-64}$$

The dc coefficient scaling factors are C_i=4S_i and the ac coefficient scaling factors are shown in Table 5.3.

YIQ VQ MBC

The coded blocks of the vector quantized YIQ MBC system use a scalar quantizer for each dc coefficient and one 9-dimensional LBG VQ for the remaining ac coefficients. If the nine ac-color components are grouped to define the vector

$$W^{(i)} = (Y_{01}^{(i,Y)}, Y_{10}^{(i,Y)}, Y_{11}^{(i,Y)}, \ldots, Y_{01}^{(i,Q)}, Y_{10}^{(i,Q)}, Y_{11}^{(i,Q)}) \tag{5-65}$$

then, four quantizers of the YIQ VQ MBC system are

$$\bar{Y}_{00}^{(i,Y)} = Q_{dc}(Y_{00}^{(i,Y)}) \tag{5-66}$$

$$\bar{Y}_{00}^{(i,j)} = C_i^{-1} Q_{ac}(C_i Y_{00}^{(i,j)}) \qquad j\varepsilon\{I,Q\} \tag{5-67}$$

$$\bar{W}^{(i)} = S_i^{-1} Q_{vlbg}(S_i W^{(i)}) \tag{5-68}$$

where the scaling factors are the same as used for YIQ SQ MBC. The LBG VQ code book used contained 1024 vectors.

YIQ SQ and VQ MBC/PT

As is the case for all of the progressive transmission coders, the YIQ SQ MBC/PT coder quantizes the first pass differently than the subsequent passes. The first pass quantizers are

similar to those of (62), (63) and (64)

$$\bar{Y}_{00}^{(1,Y)} = Q_{dc}(Y_{00}^{(1,Y)}) \tag{5-69}$$

$$\bar{Y}_{00}^{(1,j)} = Q_{ac}(Y_{00}^{(1,j)}) \qquad j\varepsilon\{I,Q\} \tag{5-70}$$

$$\bar{Y}_{kl}^{(1,j)} = S_i^{-1}Q_{ac}(S_iY_{kl}^{(1,j)}) \qquad kl = 01,\ 10 \text{ and } 11,\ j\varepsilon\{Y,I,Q\} \tag{5-71}$$

For all subsequent passes ($i>1$) all twelve quantizers are 5-OLSQ

$$\bar{Y}_{00}^{(i,j)} = S_i^{-1}Q_{ac}(S_iY_{00}^{(i,j)}) \quad j\varepsilon\{Y,I,Q\} \tag{5-72}$$

$$\bar{Y}_{kl}^{(i,j)} = S_i^{-1}Q_{ac}(S_iY_{kl}^{(i,j)}) \quad kl = 01,\ 10 \text{ and } 11,\ j\varepsilon\{Y,I,Q\} \tag{5-73}$$

All of the scaling factors for YIQ SQ MBC/PT are the same as are used for YIQ SQ and VQ MBC (Table 5.3).

The YIQ VQ MBC/PT codes the first pass, $i=1$, using three scalar quantizers for the dc coefficients and a 9-dimensional LBG VQ for the ac coefficients

$$\bar{Y}_{00}^{(1,Y)} = Q_{dc}(Y_{00}^{(1,Y)}) \tag{5-74}$$

$$\bar{Y}_{00}^{(1,j)} = Q_{ac}(Y_{00}^{(1,j)}) \qquad j\varepsilon\{I,Q\} \tag{5-75}$$

$$\bar{W}^{(1)} = S_1^{-1}Q_{9lbg}(S_1W^{(1)}) \tag{5-76}$$

All subsequent passes ($i>1$) are coded with two LBG vector quantizers. Defining a 3-dimensional vector to be made from the dc coefficient of each of the color planes

$$Y^{(i)} = (Y_{00}^{(i,Y)}, Y_{00}^{(i,I)}, Y_{00}^{(i,Q)}) \tag{5-77}$$

and a 9-dimensional vector of the ac coefficients as is shown by (65), the subsequent passes are quantized

$$\bar{Y}^{(i)} = C_i^{-1}Q_{3lbg}(C_iY^{(i)}) \tag{5-78}$$

$$\bar{W}^{(i)} = S_i^{-1}Q_{9lbg}(S_iW^{(i)}) \tag{5-79}$$

where the scaling factor are the same as are used for all of the other YIQ coders. The dc coefficient LBG VQ code book used was of size 64, and the ac coefficient code book was of size 1024.

5.3 The MBC and MBC/PT computer simulations

In this section the computer simulations that test the various MBC and MBC/PT systems of this dissertation are given. These studies include monochrome, RGB and YIQ images and are shown in Figures 5.8 through 5.26. Before moving into the specifics of the studies themselves, each is listed by coding and color type, and some general information that applies to the studies as a whole is given.

The BW MBC systems studied include four different cases:

- scalar quantized MBC only,
- scalar quantized MBC with block truncation coding,
- vector quantized MBC only and
- vector quantized MBC with block truncation coding

and the results of these studies are shown in Figures 5.8 through 5.11. The BW MBC/PT systems studied include two different cases:

- scalar quantized MBC/PT and
- vector quantized MBC/PT

and the results of these systems are shown in Figures 5.12 through 5.14.

The color MBC and MBC/PT include four cases for each color scheme

- RGB and YIQ scalar quantized MBC
- RGB and YIQ vector quantized MBC
- RGB and YIQ scalar quantized MBC/PT
- RGB and YIQ vector quantized MBC/PT

The results of these systems are shown in the following figures:

YIQ MBC	Figures 5.15 through 5.17
YIQ MBC/PT	Figures 5.18 through 5.20

RGB MBC	Figures 5.21 through 5.23
RGB MBC/PT	Figures 5.24 through 5.26

All of the above systems were tested using two different images, the woman/hat (Figure 2.27) and the F-16 (Figure 5.28). Since all of the images used in these studies, and in the quantizer design computer programs are taken from a USC database of RGB images, the BW images were taken from the G color plane of the corresponding color image, and the YIQ images were generated from the RGB images using the transformation in Pratt [3:1]. The final distortion levels of YIQ images were computed after transforming the coded image back into RGB colors.

All of these systems were tested using a largest blocksize of 16×16 and a smallest blocksize of 2×2. Therefore, each system can code images using a total of four different blocksizes (16×16, 8×8, 4×4, 2×2).

Some preliminary studies were made using 32×32 as the largest blocksize but it was found that it was were too large when coding the blocks with only four coefficients. The severe undercoding of the 32×32 blocks caused distortion to migrate over their entire pel complement, and, as a result, more of the smaller blocks had to be coded to recover from the induced distortion. The smallest blocksize can be set to 4×4 to reduce the overall coding rate without losing much in the signal-to-noise ratio, but images coded with 4×4 as the smallest blocksize suffered in perceptual quality in the regions that contained sharp edges and fine detail.

General information concerning all coders

Each of these systems was tested by adjusting the distortion thresholds from 0 to ∞. Plots were made for distortion and PSNR versus rate, and for mixture fractions versus distortion threshold. For simplicity of presentation, the thresholds were held to the same value for all the blocksizes. To adequately select the correct thresholds for the different blocksizes, it is necessary

to include perception judgement in the selection process. With the display resources available it was decided that this would be an unjustified pursuit.

Showing the performance of the coders over such a wide ranges of thresholds reveals several interesting facts. It shows how the coders work when they are not operated within their expected nominal operating range. For the vector quantized coders, the nominal operating range include a coding range from 0.5 to 1.5 bits/pel and an approximate coding fidelity of 30 dB PSNR. For the monochrome coders the nominal operating range occurs with a threshold setting of approximately 30, and for color coding at threshold setting of approximately 10.

All of the plots are tied together through the distortion thresholds. The distortion and PSNR plots listed above do not show the thresholds explicitly. This was done to keep the figures from becoming congested with threshold tics. To see how the distortion and PSNR vary as functions of threshold, consider the plots of Figures 5.5 through 5.7. These plots represent trends that are typical for all of the others.

An important point can be taken from these plots. As the threshold is tightened (set closer to 0) the rate increases (Figures 5.6 and 5.7) because the large blocks cannot pass the threshold test as easily when the threshold is tight. Therefore, the images must be coded with more small blocks. Small blocks require more bits to code, and, as a result, the rate increases. This fact is supported by the mixture percentages plots (Figure 5.4).

These plots show that as the threshold tightens the distortion decreases (the PSNR increases). The smaller blocks used to overcome the more stringent thresholds code the image with better fidelity than their large counterparts, and the distortion decreases as the number of blocks increases.

When the threshold is set to 0 the entire image is coded with the smallest blocksize. Each coded block fails the 0 distortion threshold and is partitioned into the next smallest

blocksize. This continues until all of the blocks have been divided into the smallest allowed blocksize and the coder stops. For MBC only the smallest blocksizes are coded for transmission. For MBC/PT the entire image is coded for transmission during every pass, and if four different blocksizes are used the image is transmitted four times and the image is coded at the maximum possible rate. Depending upon the coder and the quantizers used, this rate will vary. These rates are listed in Table 5.4.

When using the 0 threshold, the image is coded with the greatest number of blocks. For this case, images code with the best distortion level that can obtained (see Figures 5.6 and 5.7). On the other hand, if the threshold is set to ∞ the image is coded with only the largest blocksize. No matter how complex the original image, the ∞ threshold will always be satisfied and no blocks will be divided. Therefore, the ∞ threshold causes the image to be coded as sparingly as possible, so the rates will be low. The distortion obtained will represent the worst possible case.

Therefore, it can be seen that by coding an image at 0 and ∞ thresholds the best and worst possible coding cases are obtained. Any threshold setting that falls between these two extremes will also fall between the coding rate and coding distortion extremes they represent.

Notice that the PT coded systems are consistent with the mixture fraction prediction of equations (22). The non-zero terms of the (22b) starts at 4 (n=4 for the general coder) and decreases to 1 as the threshold increases. This is true of all of the PT coders of this chapter.

All of the vector quantizers used with these coders where designed using three training images taken from the USC database. These images are shown in Figures 5.29, 5.30 and 5.31. Now, consider the features that can be extracted from each of the different coding systems. The BW coders will be considered first.

The BW MBC results

The BW MBC systems include BTC as an option that can be used to code an image. As has been mentioned, the lowest rate that a system using BTC can obtain is 1 bit\pel. This is evident in the results collected from BW MBC and BW MBC/BTC simulations. Consider Figure 5.9. As the threshold goes to ∞ the distortion falls and the rates decrease for both systems, but the MBC/BTC rates are bounded by 1 bit/pel while the MBC rates obtain levels much lower. The MBC coders perform better at rates less than 2 to 3 bits/pel (depending upon the image coded) and the MBC/PT coders perform better for rates that are greater.

The VQ systems consistently perform better than the SQ MBC system at low rates. But, at higher rates the SQ systems may out perform the VQ systems. The VQs used were designed to work at low rates and as a result their performance falls off as the rate increases. This result is true for almost all of the VQ systems of this chapter. Notice that in the F-16 image, VQ system gains over the SQ systems are more substantial than for the woman/hat coded image.

The mixture fractions of the smaller blocks of BTC coded MBC systems (either SQ or VQ) fall more sharply as a function of threshold than do non-BTC coded systems. For a given threshold level, more 16×16 blocks pass the threshold test when BTC is used. This is consistent with the fact that the BTC systems use a one bit quantizer that is not available to the non-BTC systems.

Figure 5.32 shows (roughly) the 256×256 central portion of the 512×512 woman/hat image that has been coded using MBC. This figure is coded with:

0.549 bits/pel,

31.574 dB PSNR and

30 for the distortion threshold levels.

The mixture fractions for the different blocksizes are:

48.73% 16×16 blocks,

25.93% 8×8 blocks,

19.59% 4×4 blocks and

5.75% 2×2 blocks.

This image is coded with a compression ratio of ~15:1 (8/.549) and even though the PSNR is good for this compression ratio it is possible to see blocking in the image. The device used to print the figure has only 32 gray scales intensity and false contouring can be seen. Some of the contours that are seen in the image are not visible when it is displayed with 8 bits of resolution.

Figure 5.33 shows the blocksize distribution that produced the woman/hat MBC image of Figure 5.32. The larger the block, the more darkly it is shaded in the figure. The 2×2 blocks indicate the image regions that are the most difficult to code. These occur at the edge of the hat, in the feathers and eyes. The 16×16 blocks show the image regions that are easy to code because of their lower busyness level; the cheek and forehead.

Figures 5.34 and 5.35 show the central portion of the F-16 image that are coded using MBC and MBC/BTC, respectively. The images were coded to give the same distortion levels. The MBC image is coded with:

0.735 bits/pel,

32.267 dB PSNR and

30 for the distortion threshold levels.

and the mixture fractions for the different blocksizes are:

48.73% 16×16 blocks,

19.17% 8×8 blocks,

21.17% 4×4 blocks and

10.13% 2×2 blocks.

The MBC/BTC image is coded with:

1.452 bits/pel (1.118 bits/pel for BTC requirements),

32.387 dB PSNR and

33 for the distortion threshold levels.

and the mixture fractions for the different blocksizes are:

57.91% 16×16 blocks,

23.97% 8×8 blocks,

16.91% 4×4 blocks and

1.20% 2×2 blocks.

Notice that even though the distortion levels are nearly the same for both images, the MBC/BTC version requires significantly more bits to code. Since the MBC/BTC version uses the additional BTC quantizer it has data that is not available in the non-BTC MBC version. Therefore, it can code each block more accurately and needs fewer smaller blocks to the same distortion levels that are obtained when using the non-BTC version.

An important feature of the BTC image is that the background contouring (or smoothness as is the case for the 8 bit coded image) of the non-BTC image is not seen. This results from the one bit quantizer "jumping" from level to level as it codes the image.

The blocksize distributions of these images is shown in Figures 5.36 for MBC, and Figure 5.37 for MBC/BTC. As is indicated by the mixture fraction lists, these figures show that the BTC version of the image is coded with far fewer small blocks. The small blocks that are needed to code the high contrast regions when using MBC system are not needed with the MBC/BTC system.

Even though both coding systems produce equal results for image areas that contain high contrast, the BTC version shows small detail in the other image regions with greater perceptual clarity. Thus, SNR (PSNR for these images) is not a good indicator of overall image

perceptual quality.

The BW MBC/PT results

The MBC/PT coders do not use BTC since progressive transmission requires that the same blocks must be coded more than once, and the need to use a 1 bit quantizer for each pel of each of these passes will increase the rate of such systems to an unusable level very quickly. This point precluded BTC from being used with MBC/PT.

The main point to be taken from Figure 5.13 is that the SQ MBC/PT system used to code the F-16 image does not overtake the performance level of the VQ system at high rates. This is different than the result obtained when these coders were used to code the woman/hat image.

Figures 5.38 through 5.42 show the four different transmission passes of the woman/hat code using the MBC/PT system. Figure 5.38 shows the image as presented after the 16×16 coding pass, and Figure 5.41 shows the image after the final 2×2 coding pass. The final image is coded with:

0.558 bits/pel,

31.599 dB PSNR and

30 for the distortion threshold levels.

The mixture fractions and accumulative coding rate each pass are:

100.00%	.034	bits/pel	16×16 block pass,
51.27%	.107	"	8×8 blocks,
24.24%	.319	"	4×4 blocks and
5.14%	.549	"	2×2 blocks.

This image is coded with a final compression ratio and PSNR that is nearly identical to that of the MBC coded version. If the 512×512 image were transmitted over a 9600 baud line, the times required to send each pass are:

0.93 seconds 16×16 blocks,

2.92 " 8×8 blocks,

8.71 " 4×4 blocks and

15.0 " 2×2 blocks.

The observer would have the first pass image in less than a second and the final image in 15 seconds.

If the image were been presented through an interactive telebrowsing system the user would be able to decide if the entire image was to be kept or rejected after about 1 second of transmission time. Once the image is recognized it can be rejected if it is was not wanted, instead of waiting the entire 15 seconds for the final image to be constructed.

Figure 5.42 shows the blocksize distribution that produced the woman/hat MBC/PT image. It is similar to the block distribution of the MBC image.

The color coder results

The most important point that is brought out by the figures of data is that most of the systems will code a color image with good PSNR (>30 dB) for rates as low as 0.32 ($10^{-.5}$ on the plots) bits/pel. The color images coded with low rate have been coded with a compression ratio of 75:1.

Since the results obtained for the color MBC and MBC/PT systems parallel the result of the BW systems nothing will be repeated that has been said before. The most important feature of concern is found with the RGB MBC/PT systems. Even though the VQ version out performs the SQ version at very low rates, its performance falls off dramatically as the rate increases. The PSNR never goes above 30 dB for either image. This may be caused by a poorly selected training set.

Final remarks concerning the computer simulations

It is possible to change the coding characteristics of MBC and MBC/PT systems by changing the block coders and the distortion thresholds used to obtain good coding result for both low- and medium-rate source coders. The use of vector quantizers in these coders is of the most advantage at low coding rates where they out perform their scalar quantized counterparts to the greatest degree.

Since the block coder of any different blocksize can be designed independently of the MBC and MBC/PT algorithm flow, many different systems can be constructed. Each of these differing systems will have coding rates where they are of the most use, and by combining different systems it is possible to code images over a wide ranges of rates.

Examples to demonstrate the MBC and MBC/PT methods where given in this chapter. The particular source coders tested used a four coefficient DCT coding scheme for each of the four blocksizes used. The coders were constructed to use 16×16 as the largest blocksize and 2×2 as the smallest. Both scalar and vector quantizers were used to code the DCT coefficients, and the special design problems associated with the design and use of the different quantizers was discussed. A new block truncation coding method was developed and used with the DCT MBC monochrome coders.

All of the coders tested where constructed with a quad tree structure and the number of blocks coded at each level of the tree were chosen with a distortion threshold. Two different distortion measures were used, one for monochrome images and another for color images. How the distortion and coding rate vary with the threshold level was also demonstrated. The examples of this chapter include coding rates from less than 0.1 bits/pel to about 8 bits/pel.

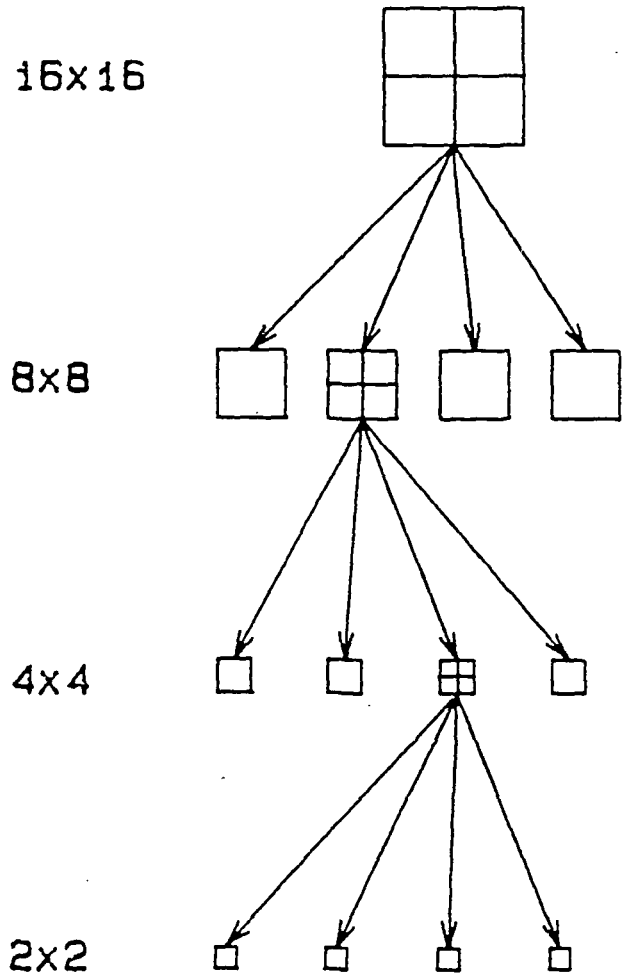


Figure 5.1 Example 16x16 block quad tree.

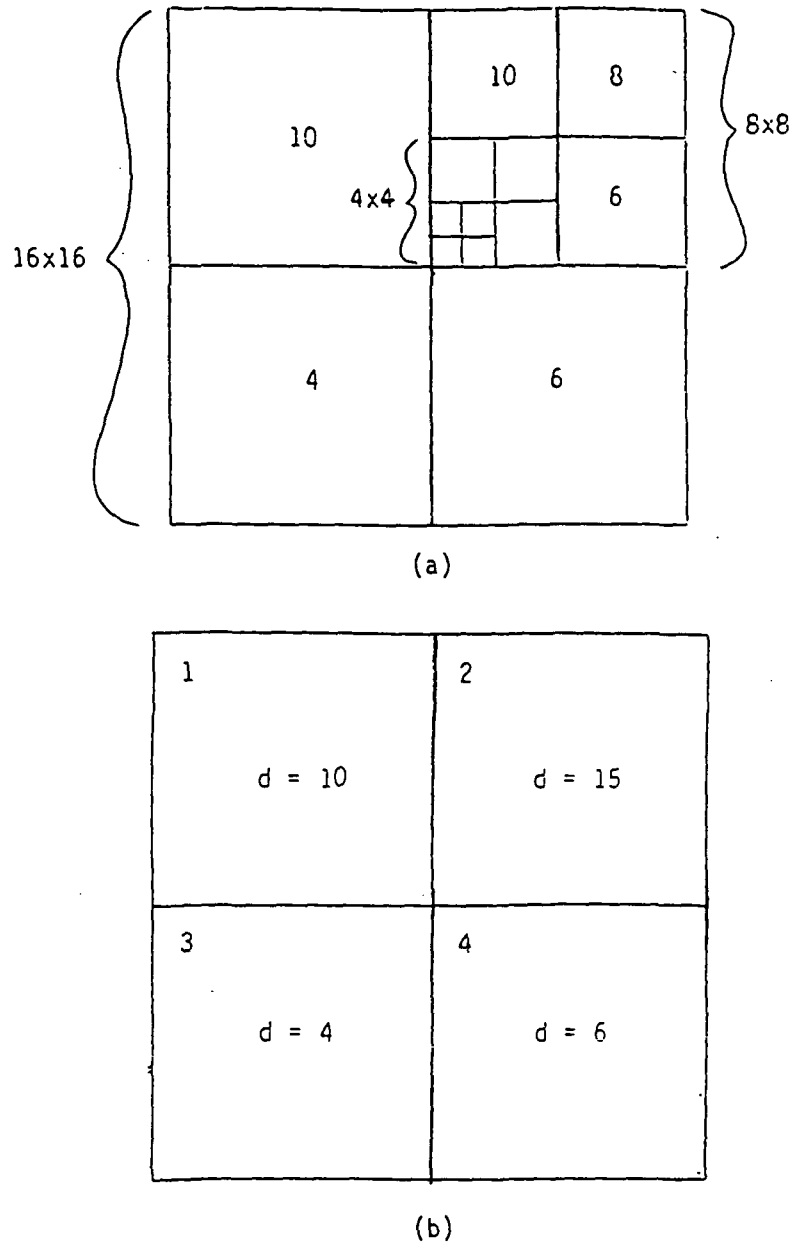


Figure 5.2 Example 16x16 block for MBC and default sub-block numbering for MBC.

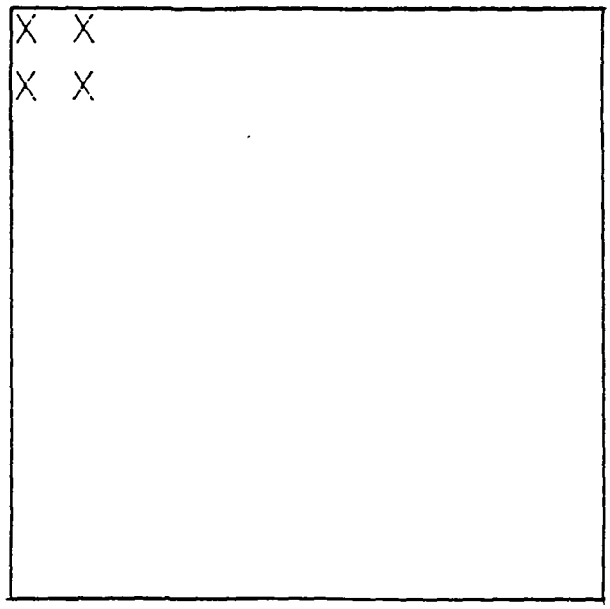


Figure 5.3 MBC and MBC/PT DCT transform coefficients.

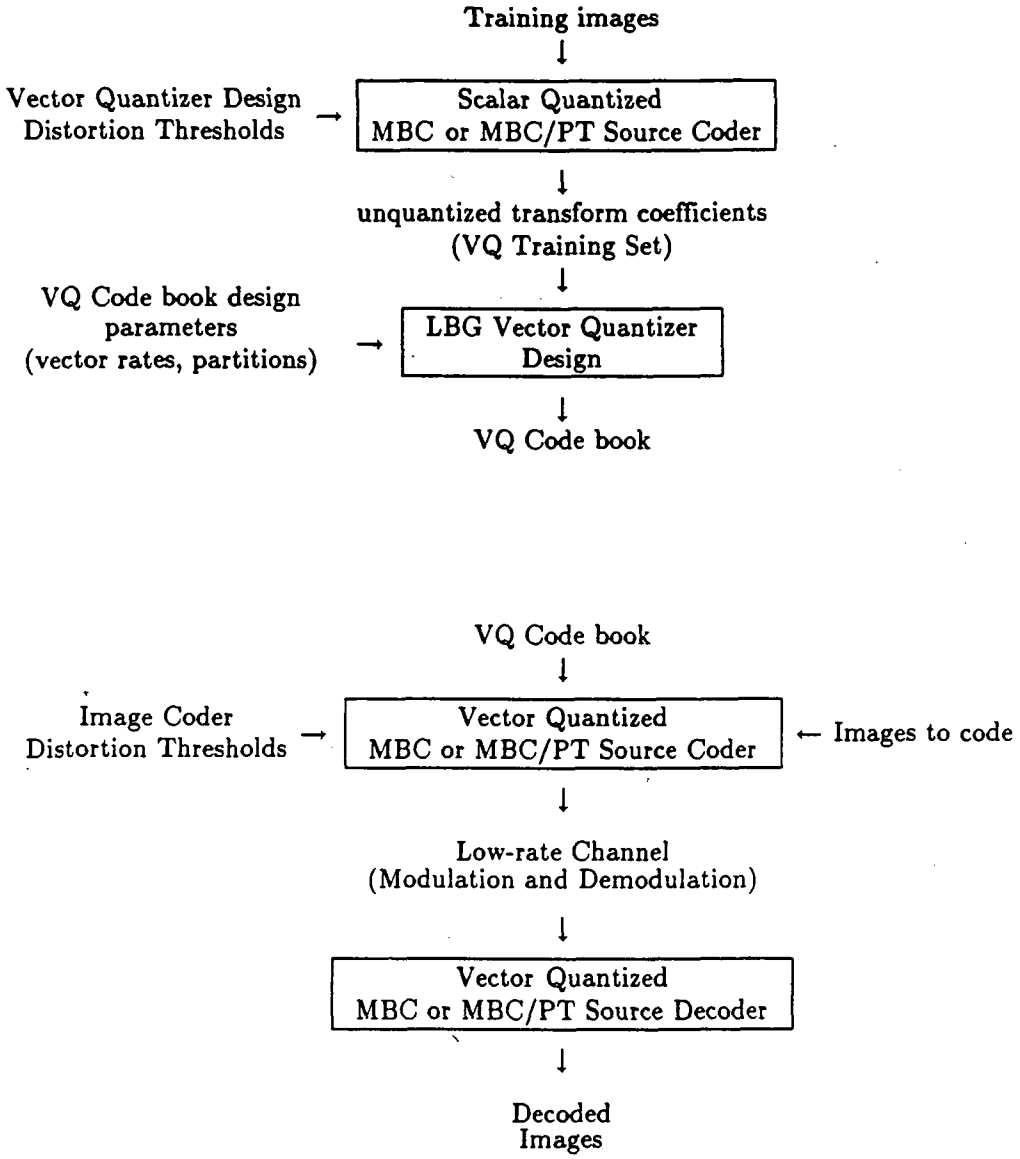


Figure 5.4 Design and use of the MBC and MBC/PT vector quantizer.

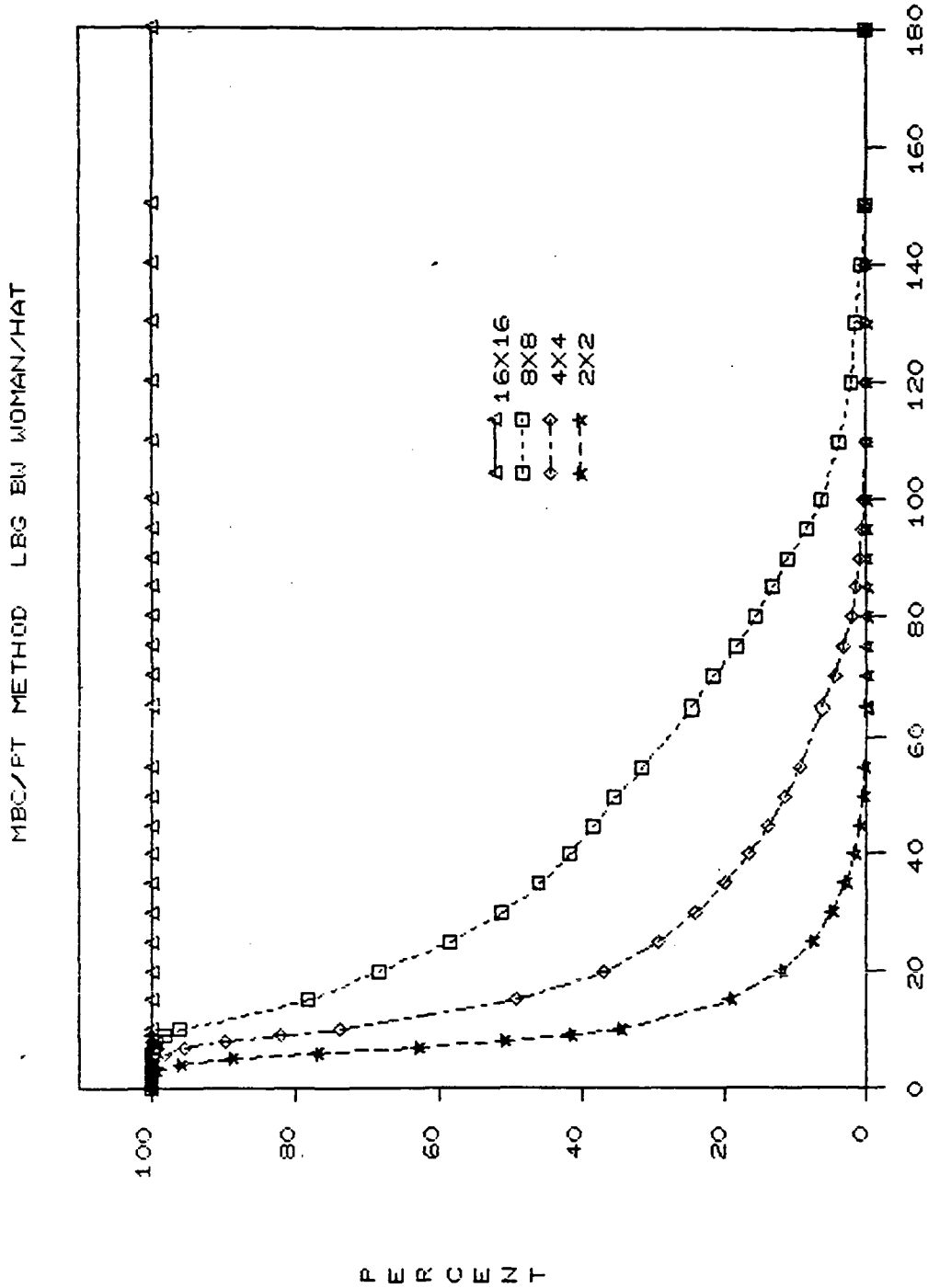


Figure 5.5 Mixture fractions versus blocksize-constant distortion thresholds.

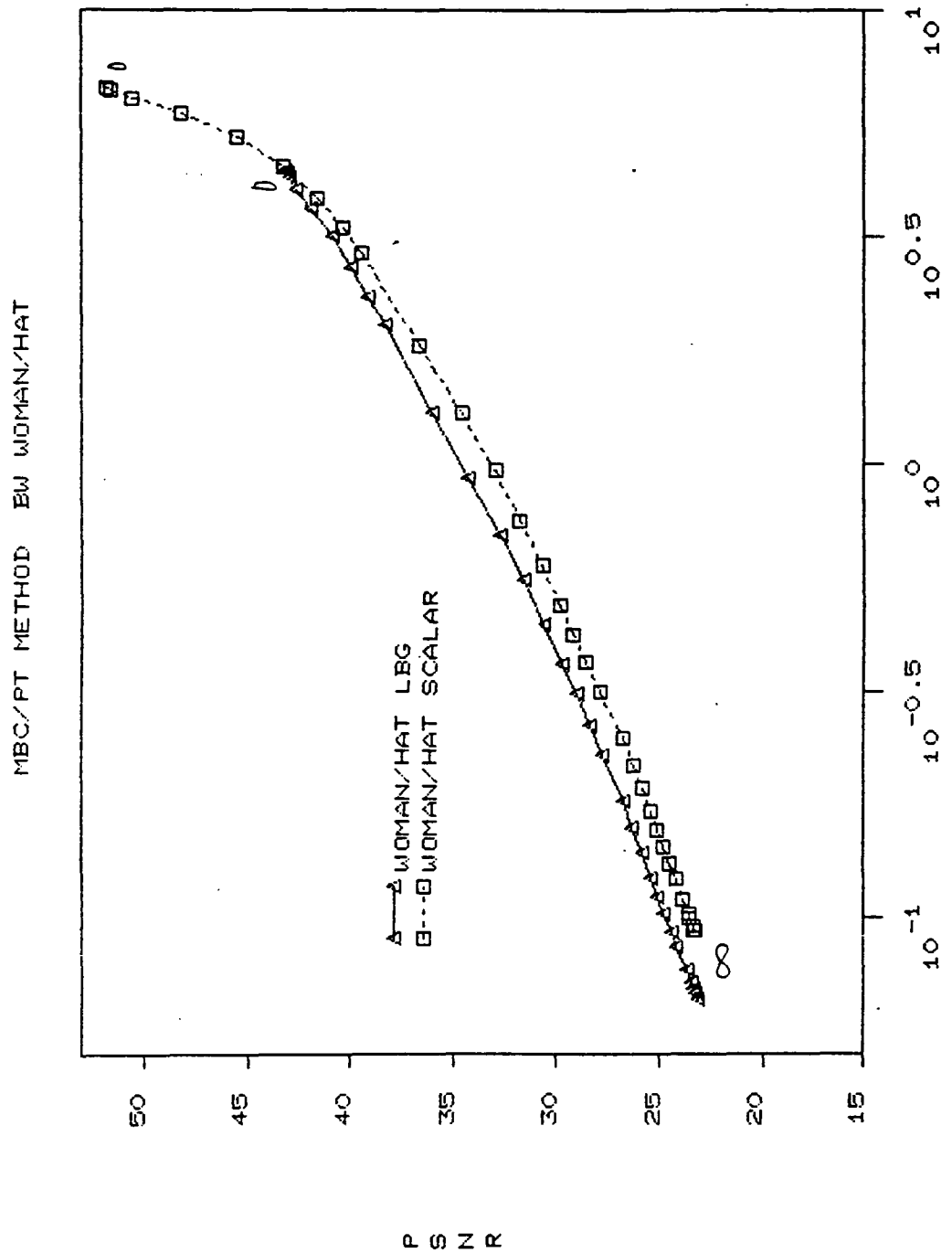


Figure 5.6 PSNR versus rate as a function of distortion threshold.

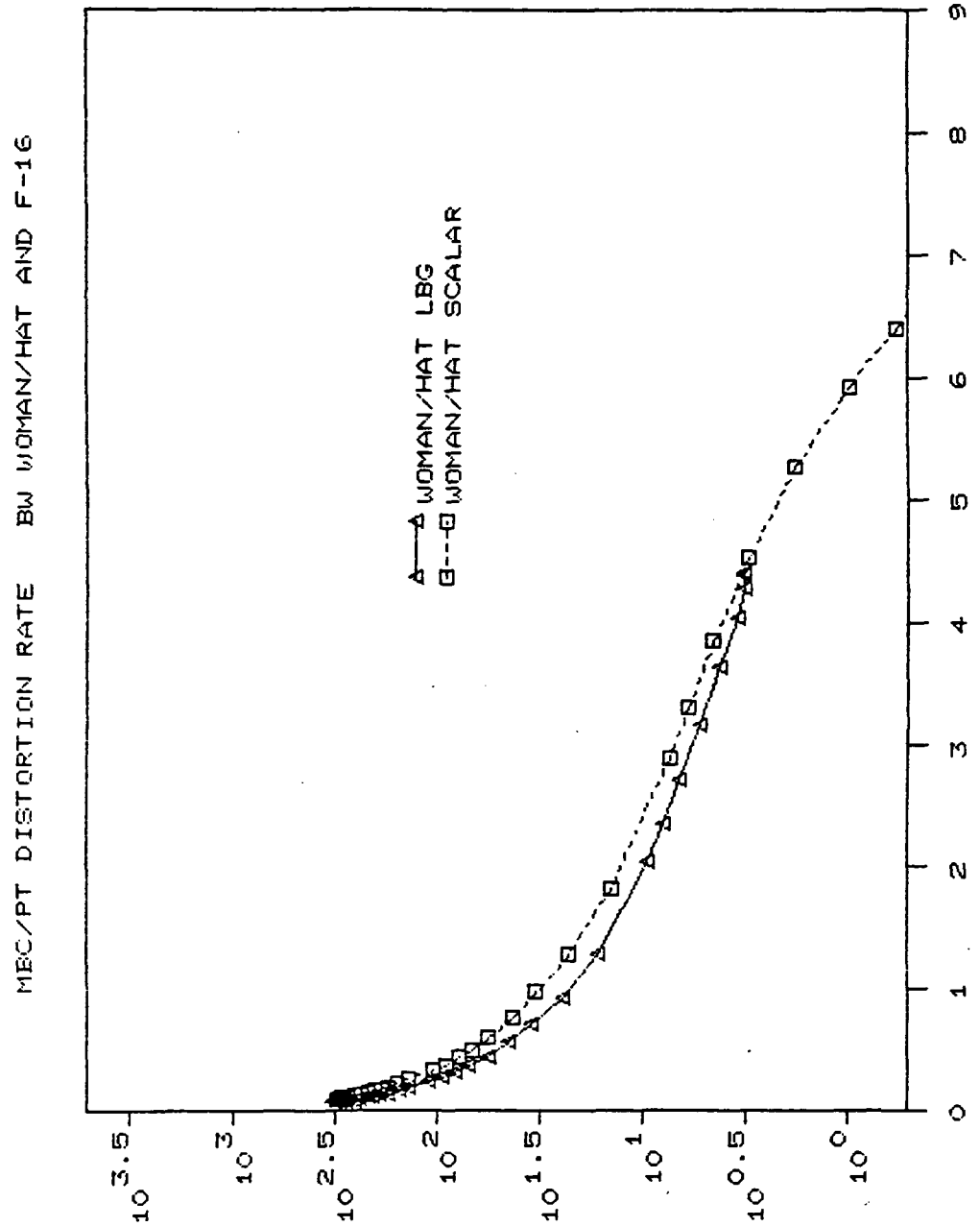


Figure 5.7 Distortion versus rate as a function of distortion threshold.

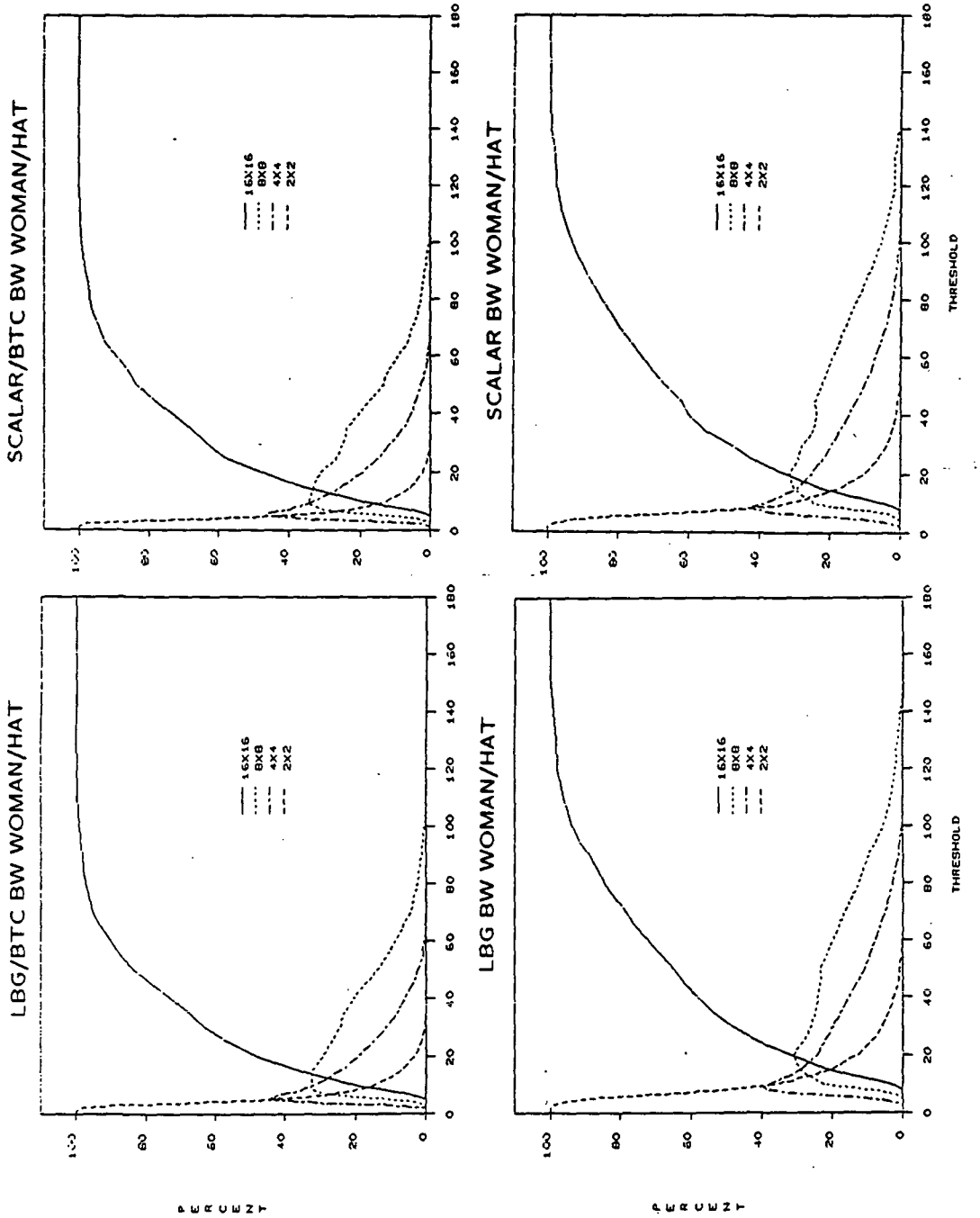


Figure 5.8 MBC mixture fractions versus threshold for BW woman/hat.

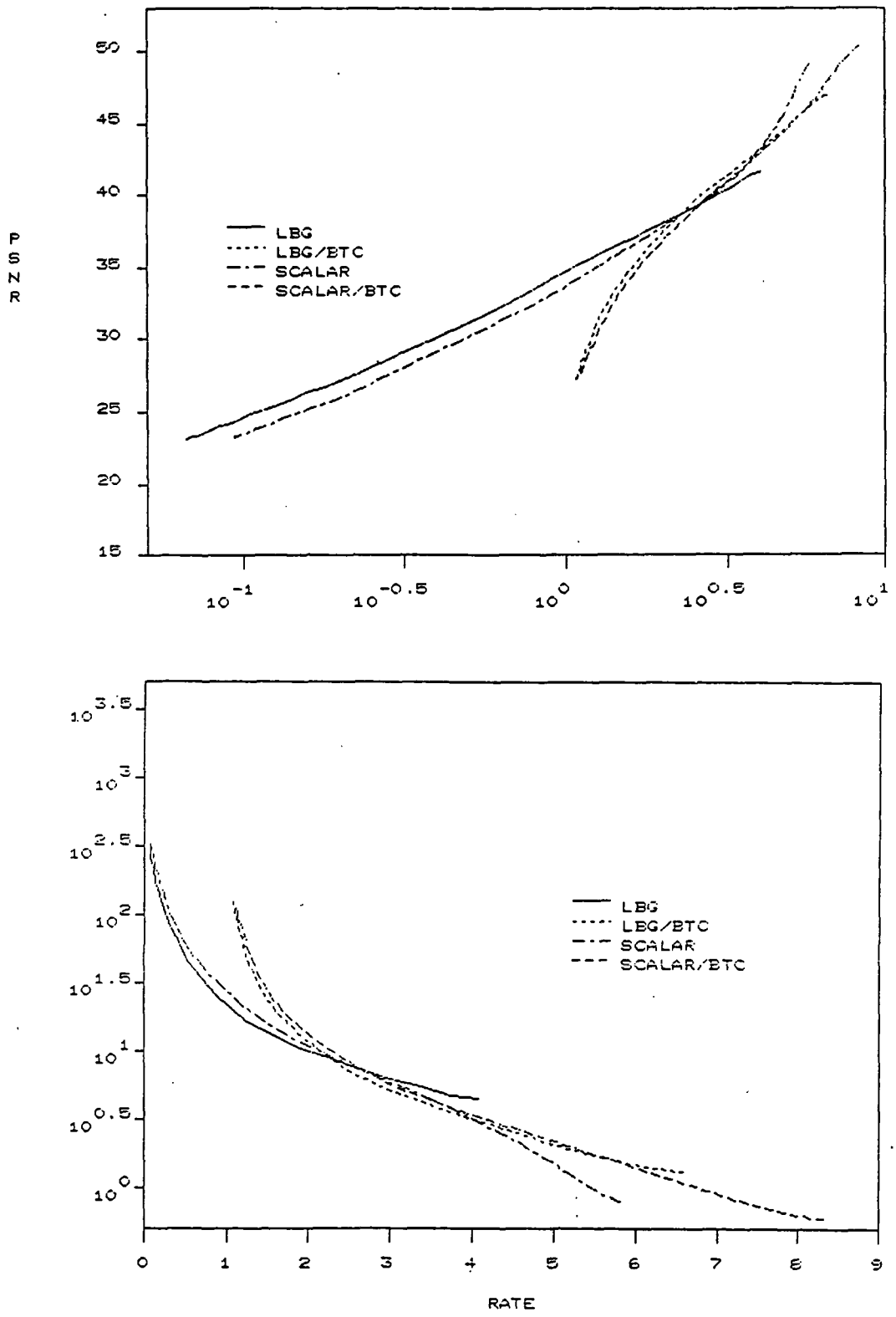


Figure 5.9 MBC PSNR and distortion versus rate for BW woman/hat.

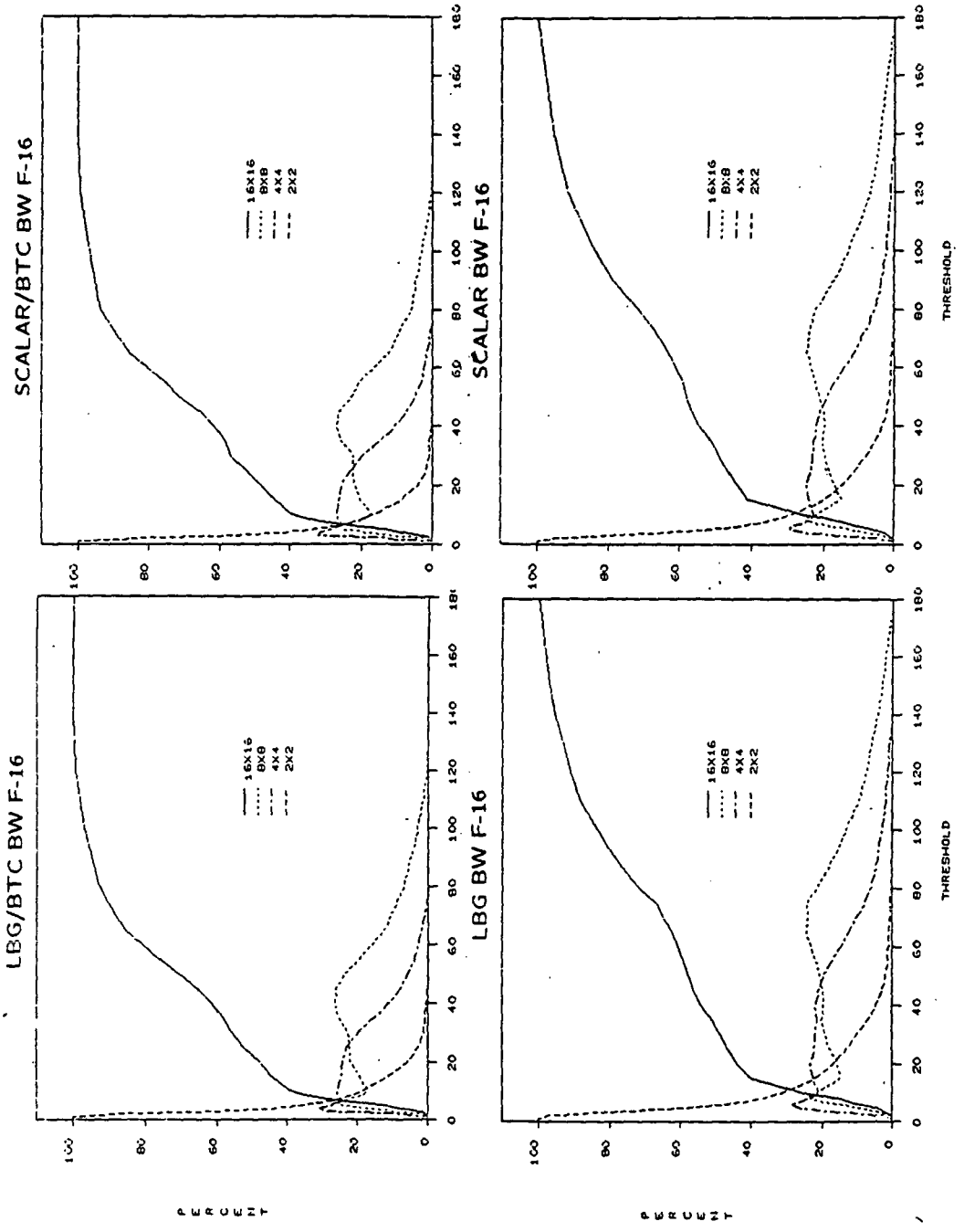


Figure 5.10 MBC mixture fractions versus threshold for BW F-16.

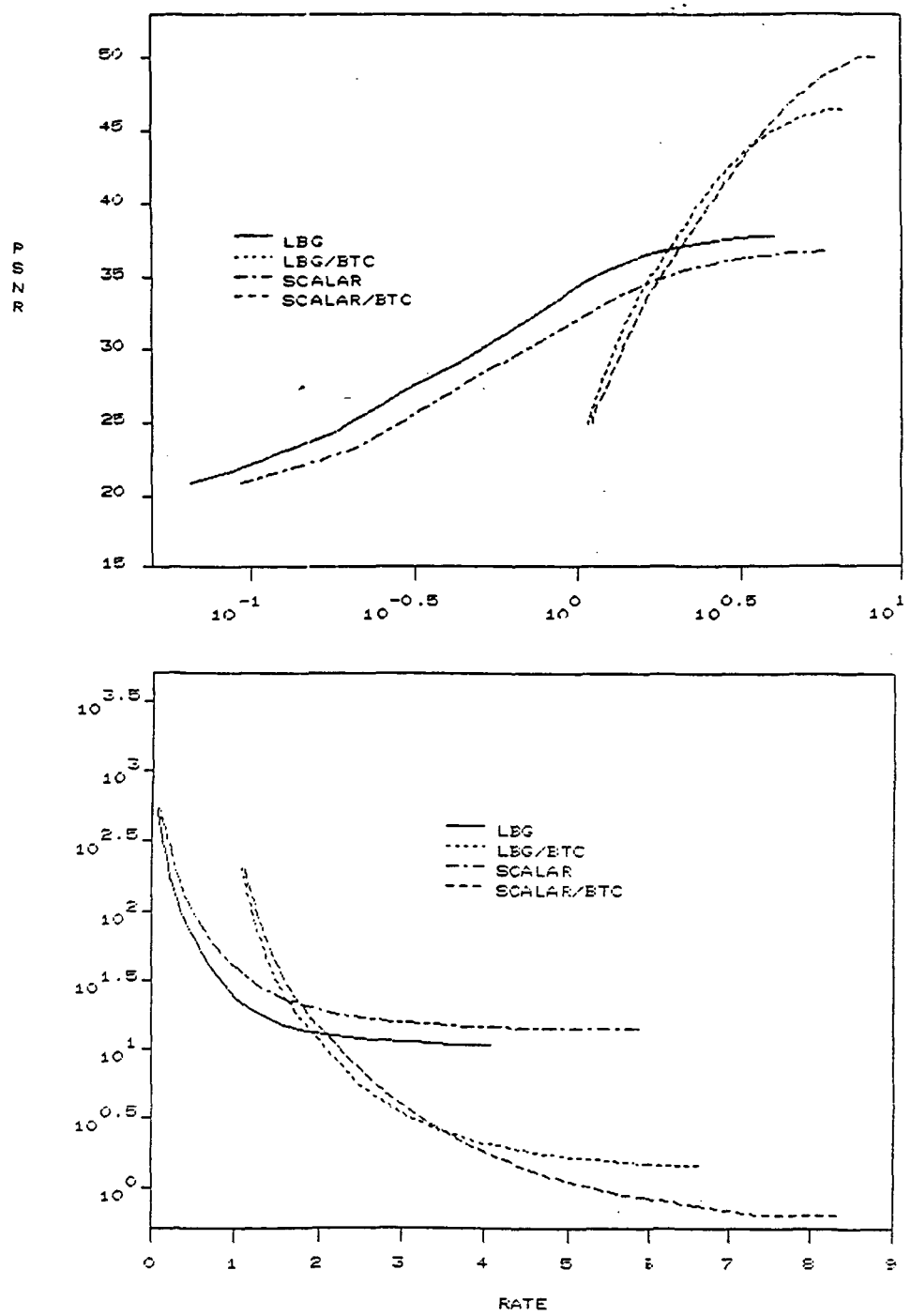


Figure 5.11 MBC PSNR and distortion versus rate for BW F-16.

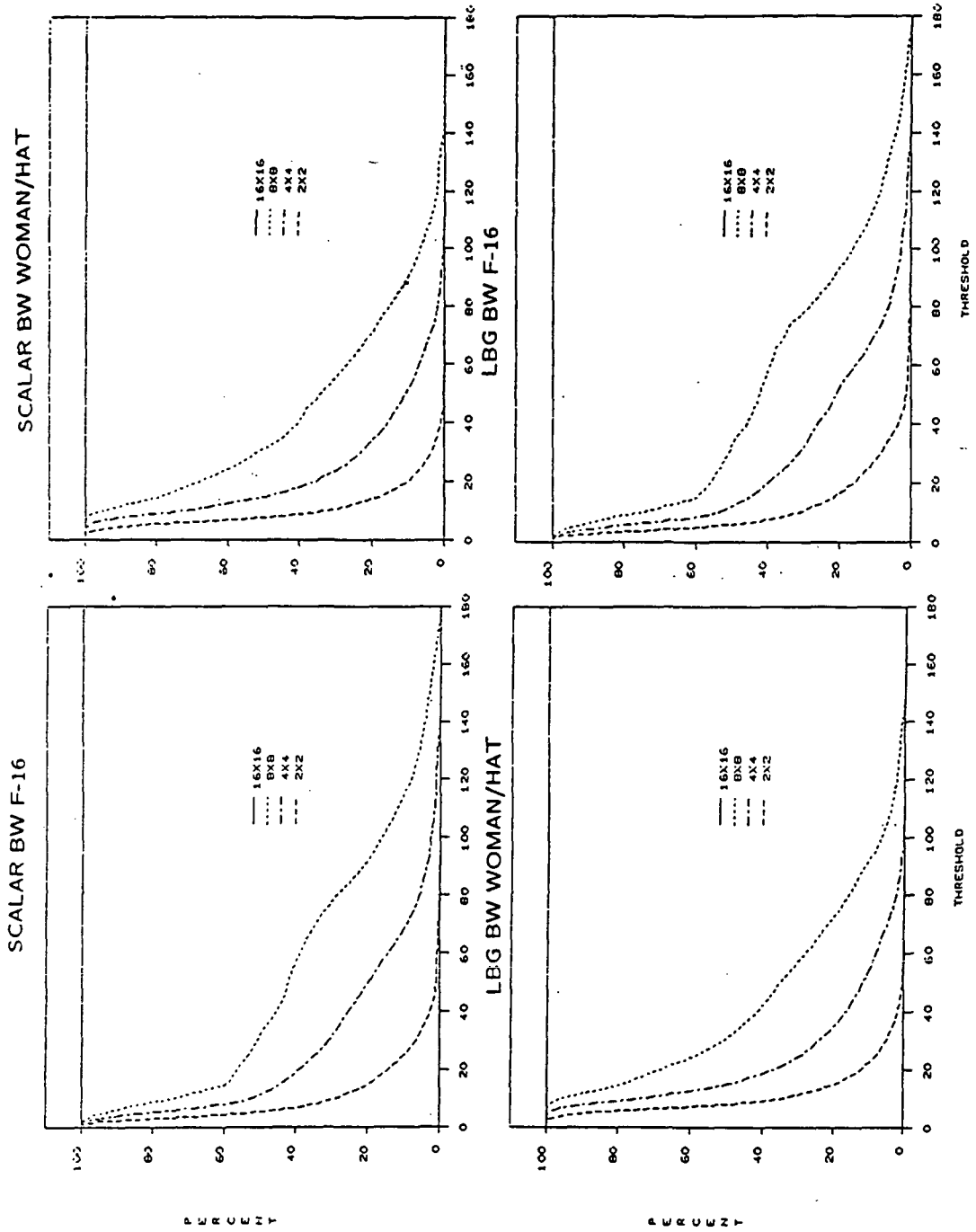


Figure 5.12 MBC/PT mixture fractions versus threshold for BW woman/hat and F-16.

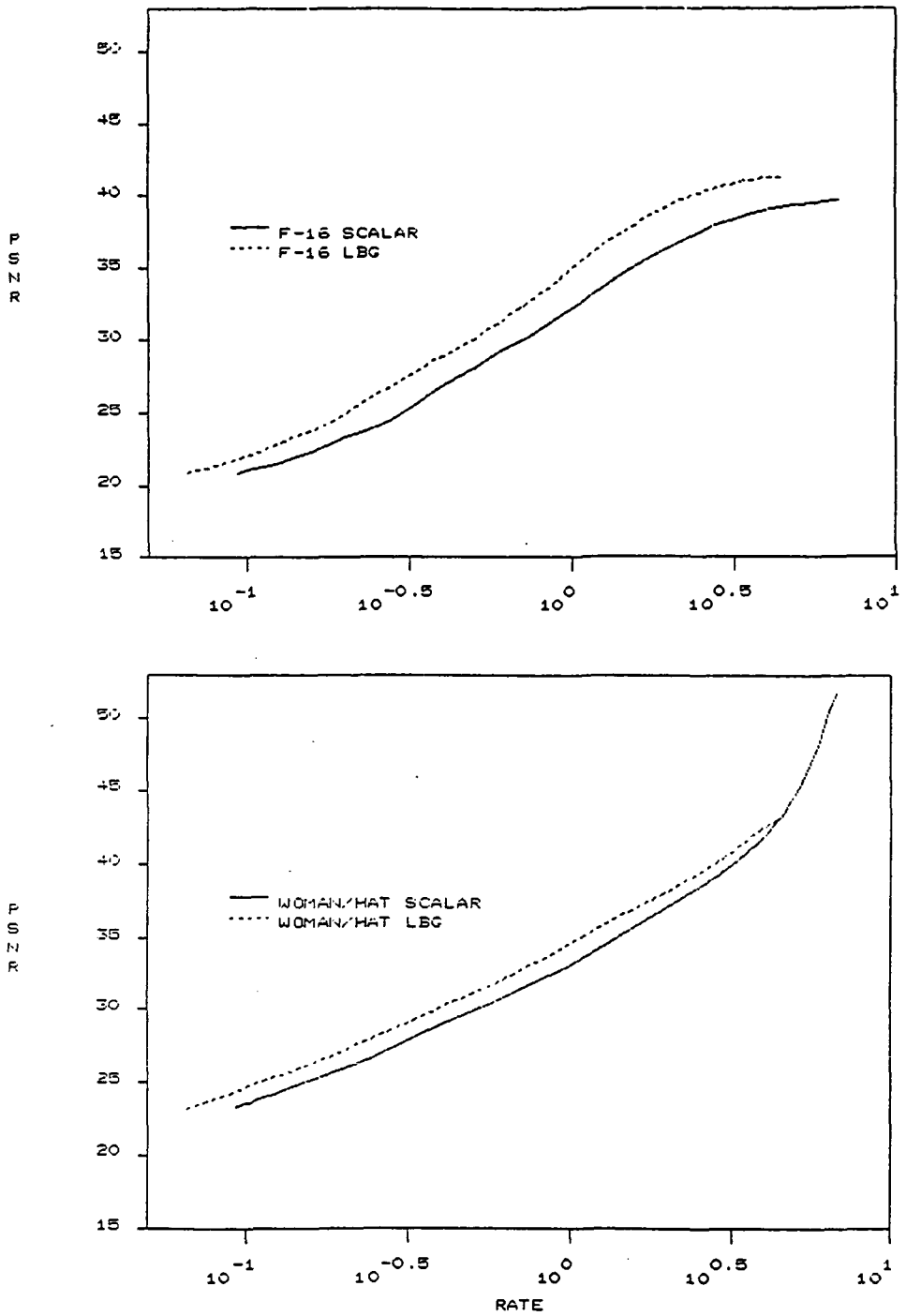


Figure 5.13 MBC/PT PSNR versus rate for BW woman/hat and F-16.

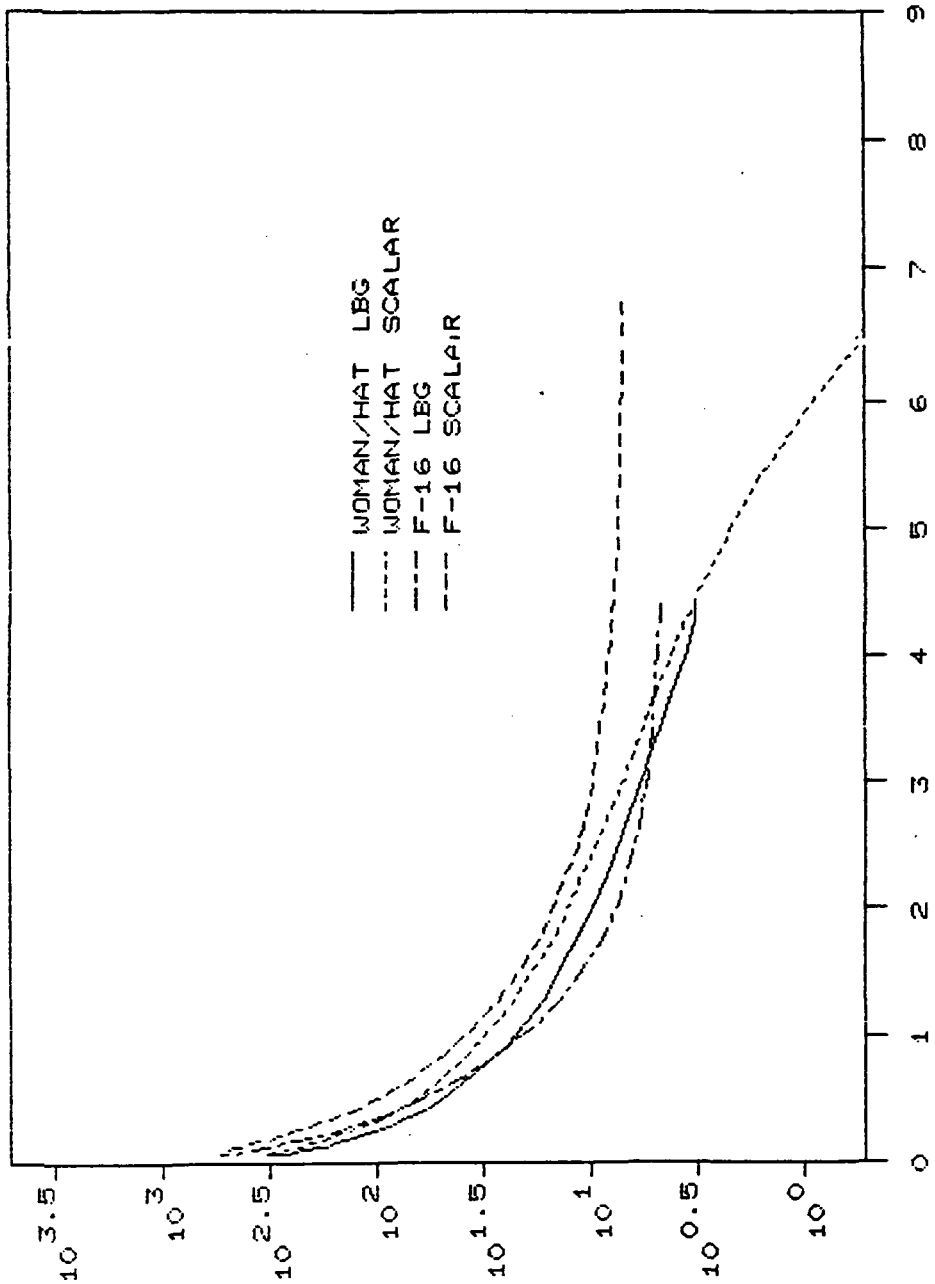


Figure 5.14 MBC/PT distortion versus rate for BW woman/hat and F-16.

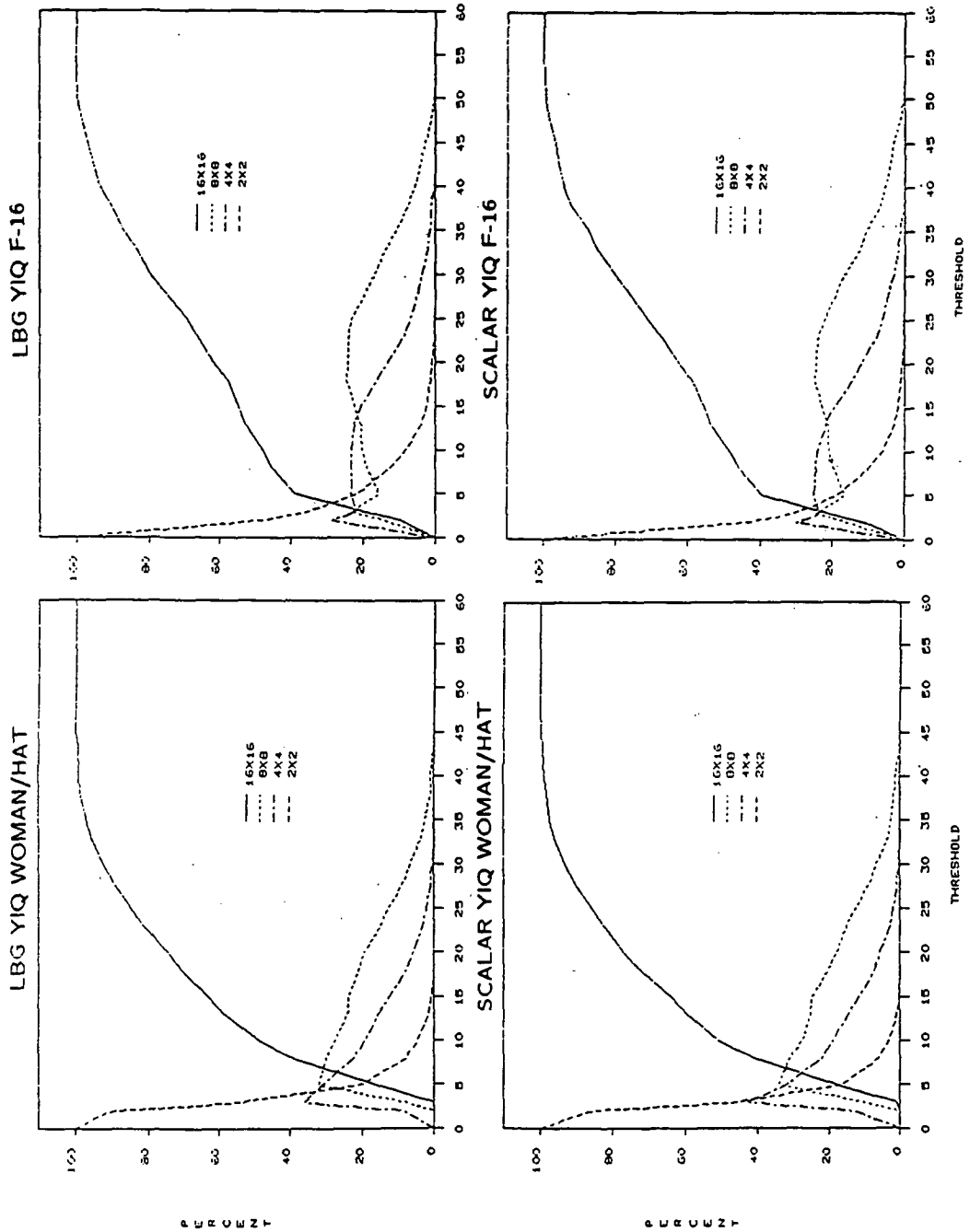


Figure 5.15 MBC mixture fractions versus threshold for YIQ woman/hat and F-16.

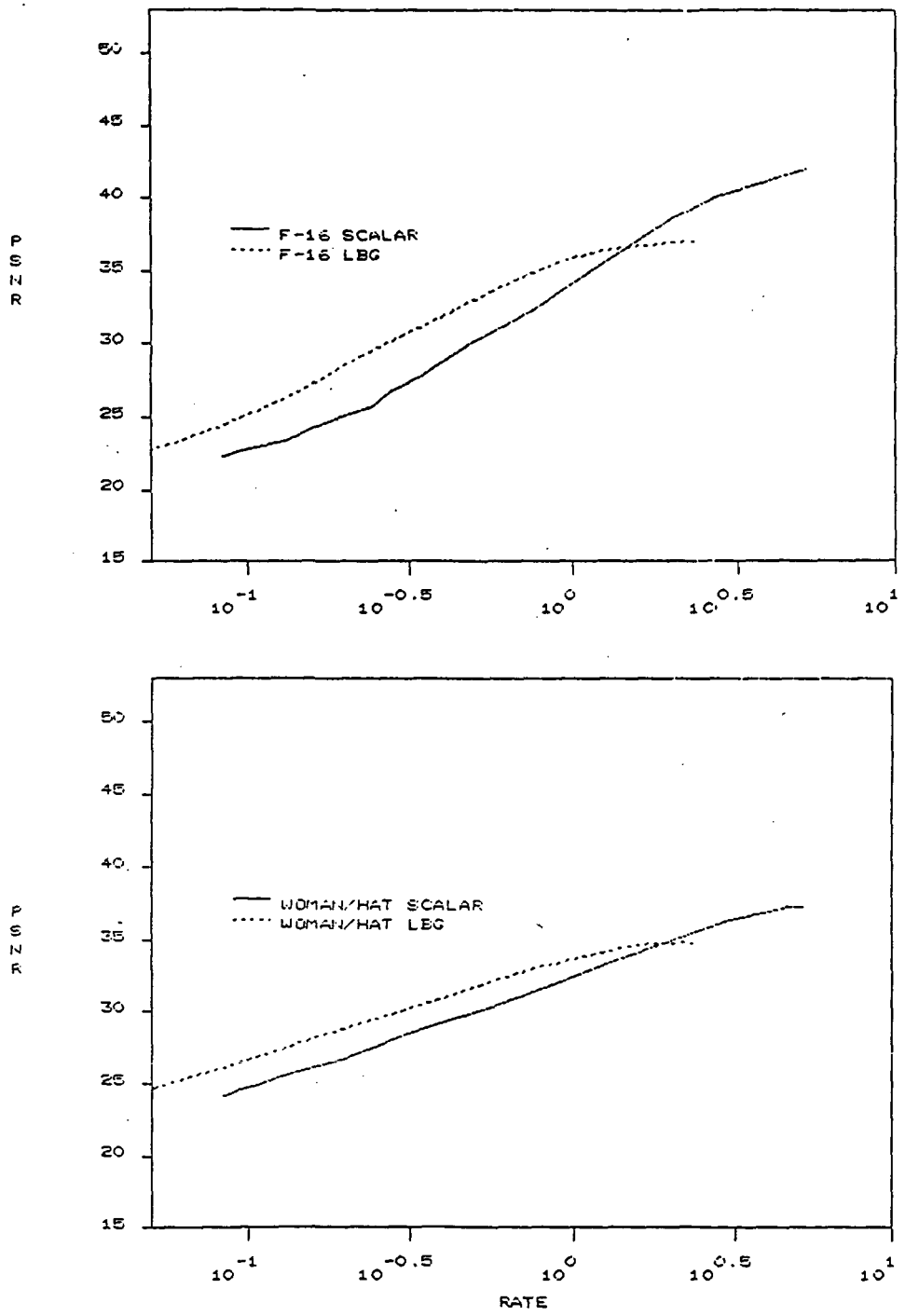


Figure 5.16 MBC PSNR versus rate for YIQ woman/hat and F-16.

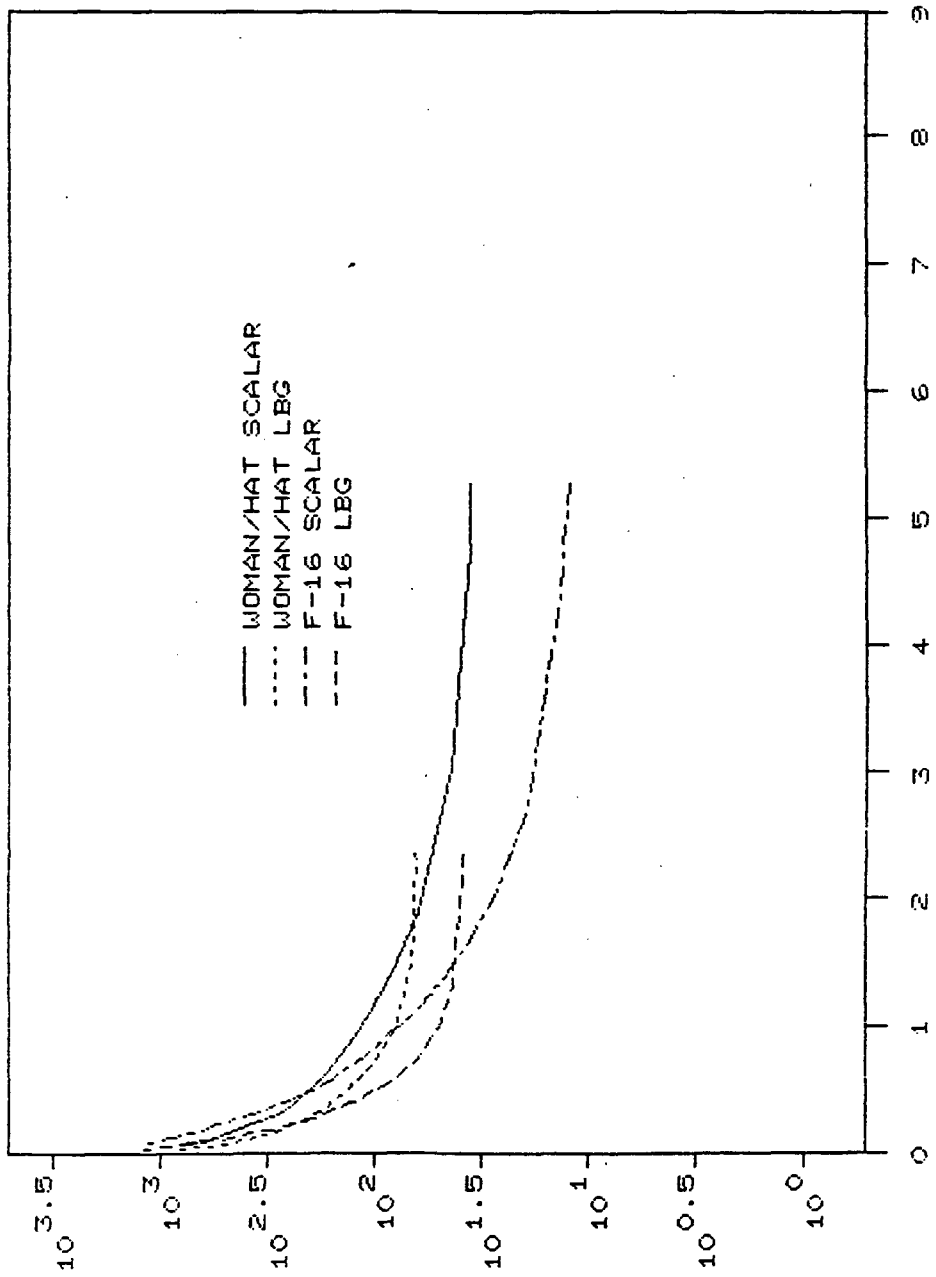


Figure 5.17 MBC distortion versus rate for YIQ woman/hat and F-16.

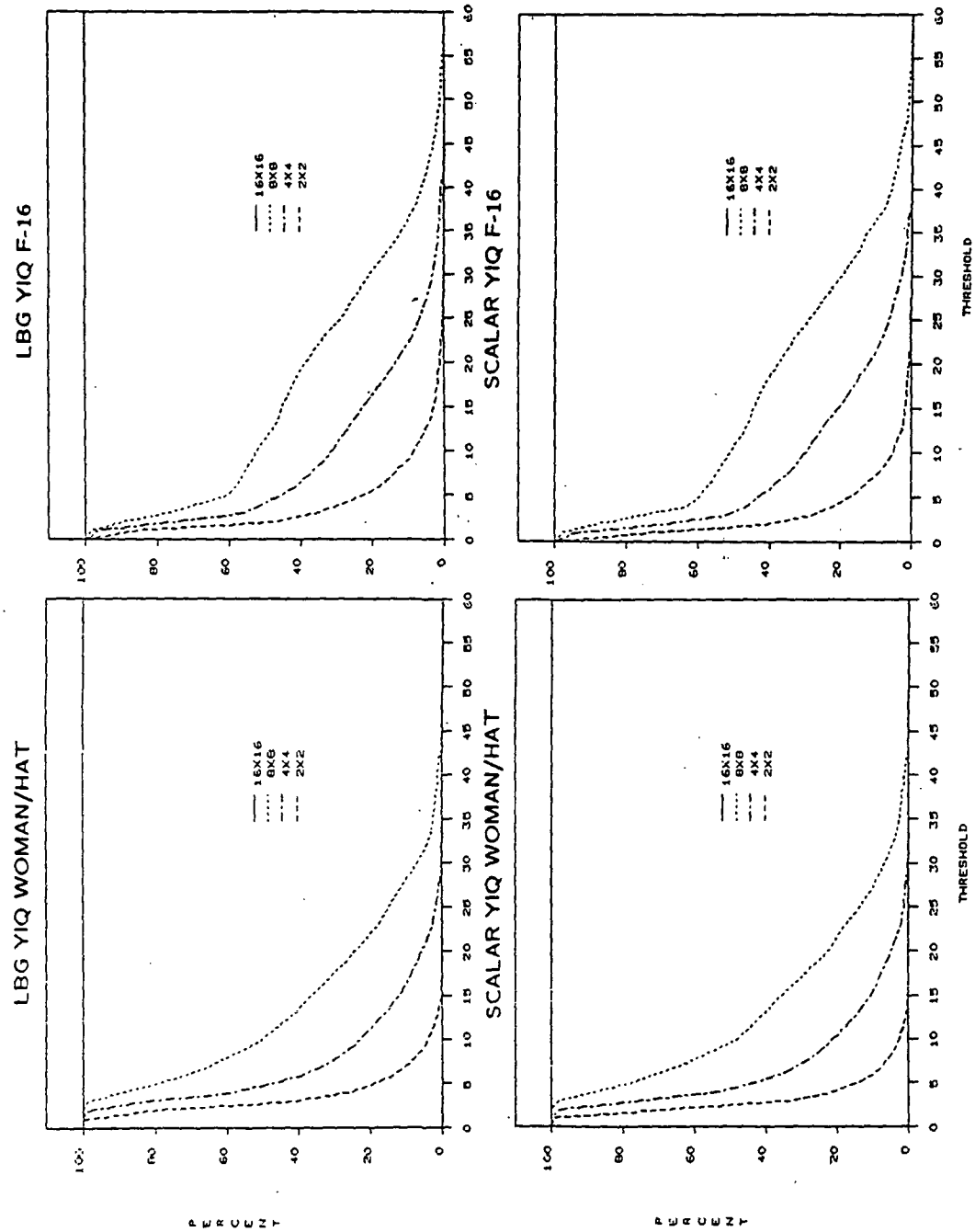


Figure 5.18 MBC/PT mixture fractions versus threshold for YIQ woman/hat and F-16.

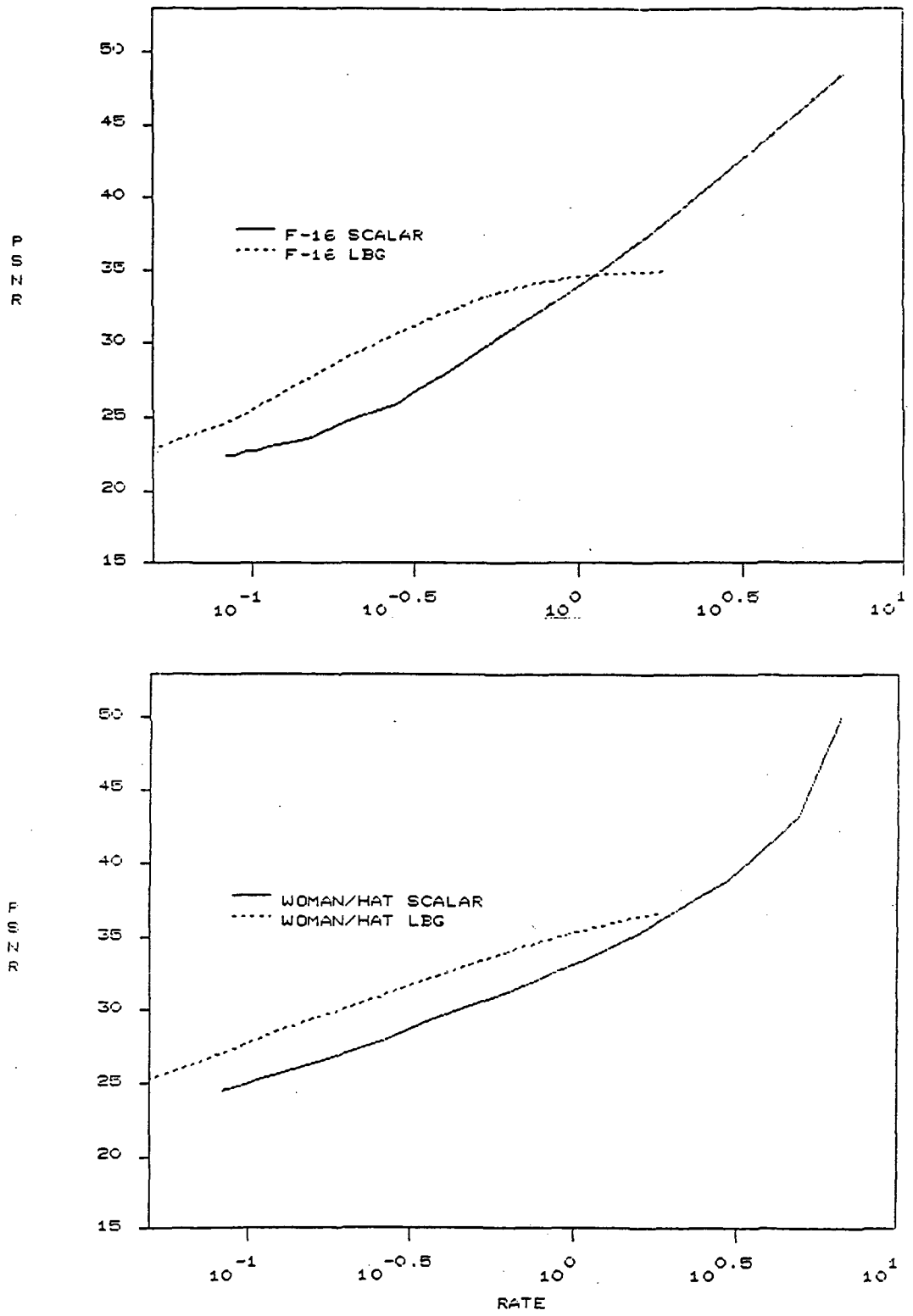


Figure 5.19 MBC/PT PSNR versus rate for YIQ woman/hat and F-16.

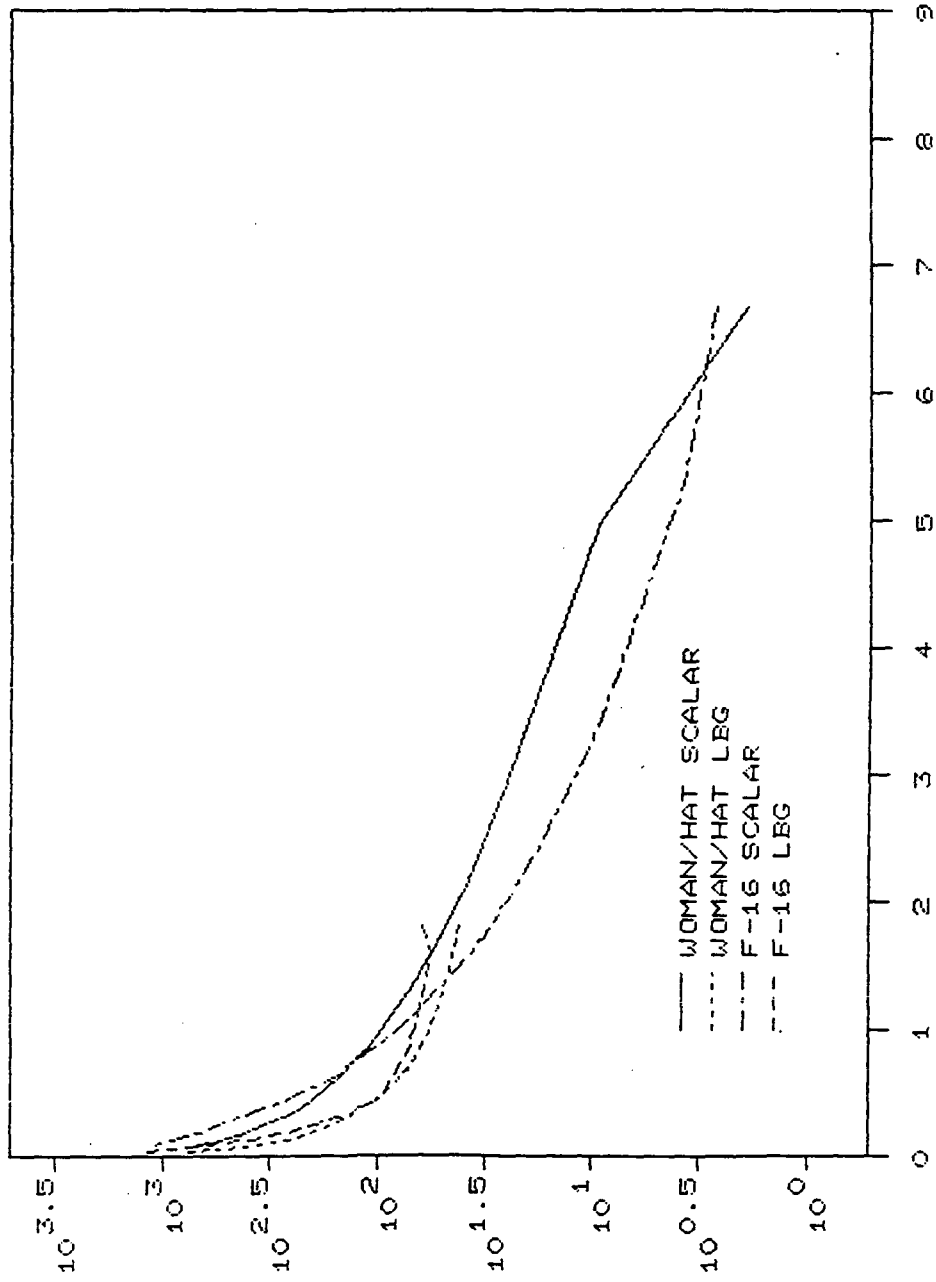


Figure 5.20 MBC/PT distortion versus rate for YIQ woman/hat and F-16.

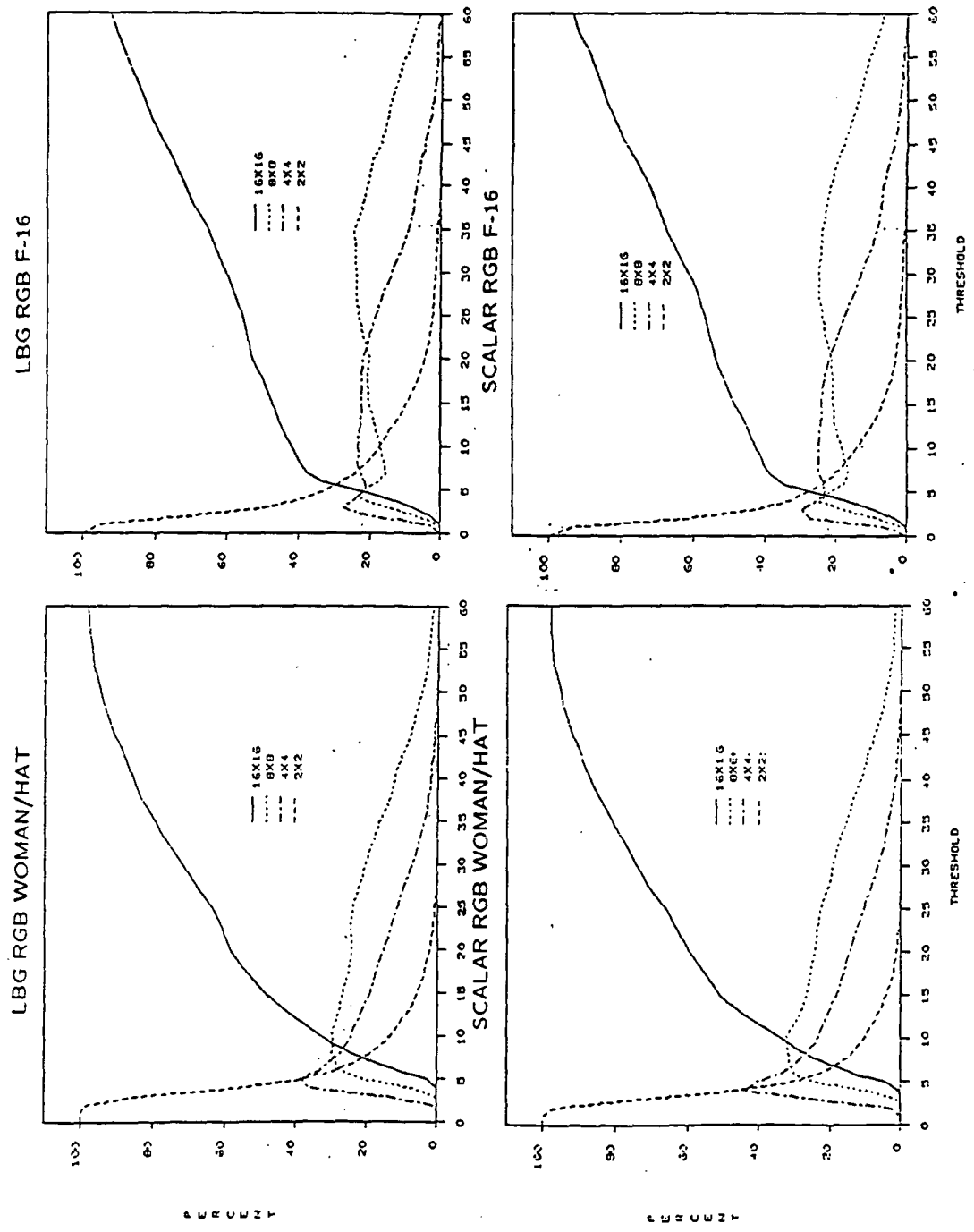


Figure 5.21 MBC mixture fractions versus threshold for RGB woman/hat and F-16.

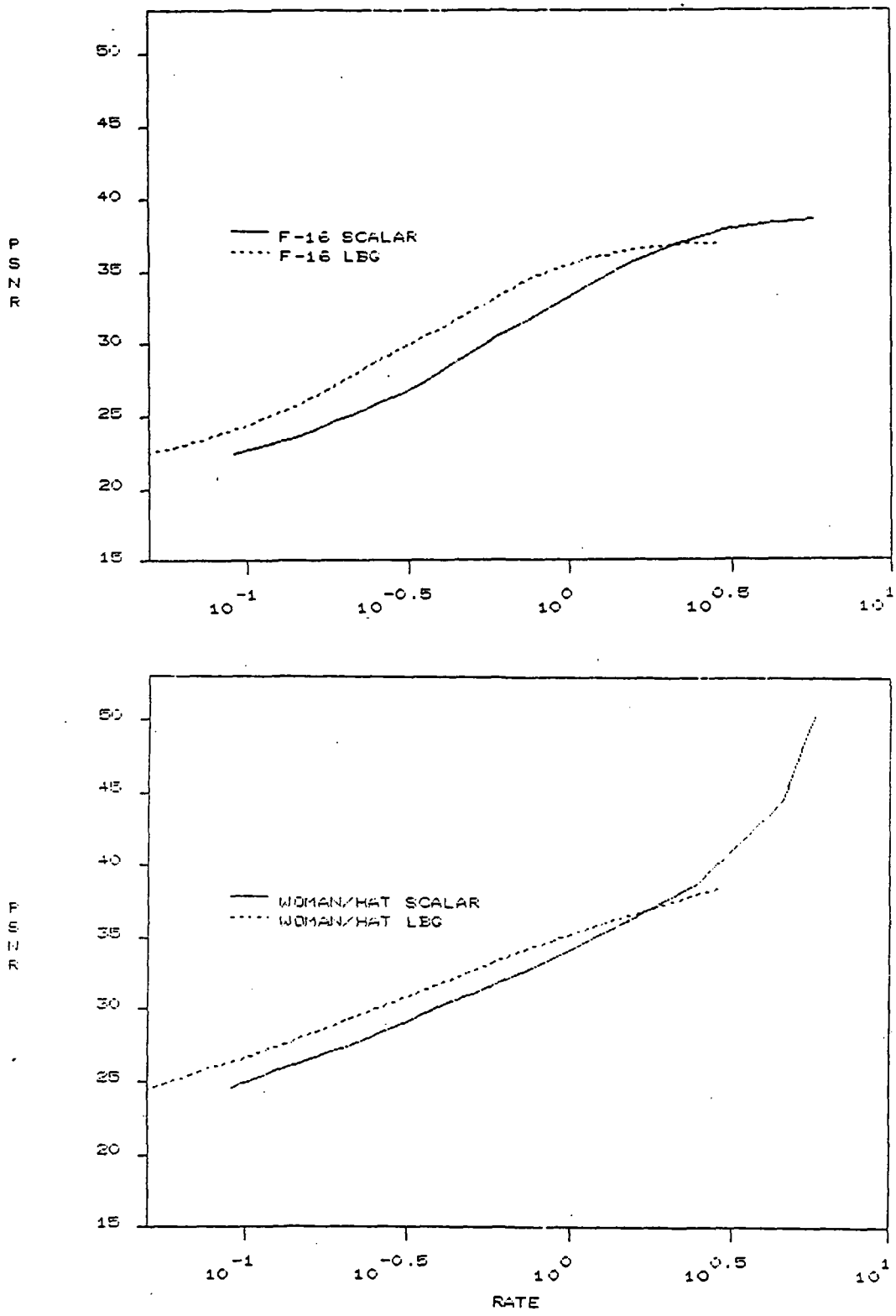


Figure 5.22 MBC PSNR versus rate for RGB woman/hat and F-16.

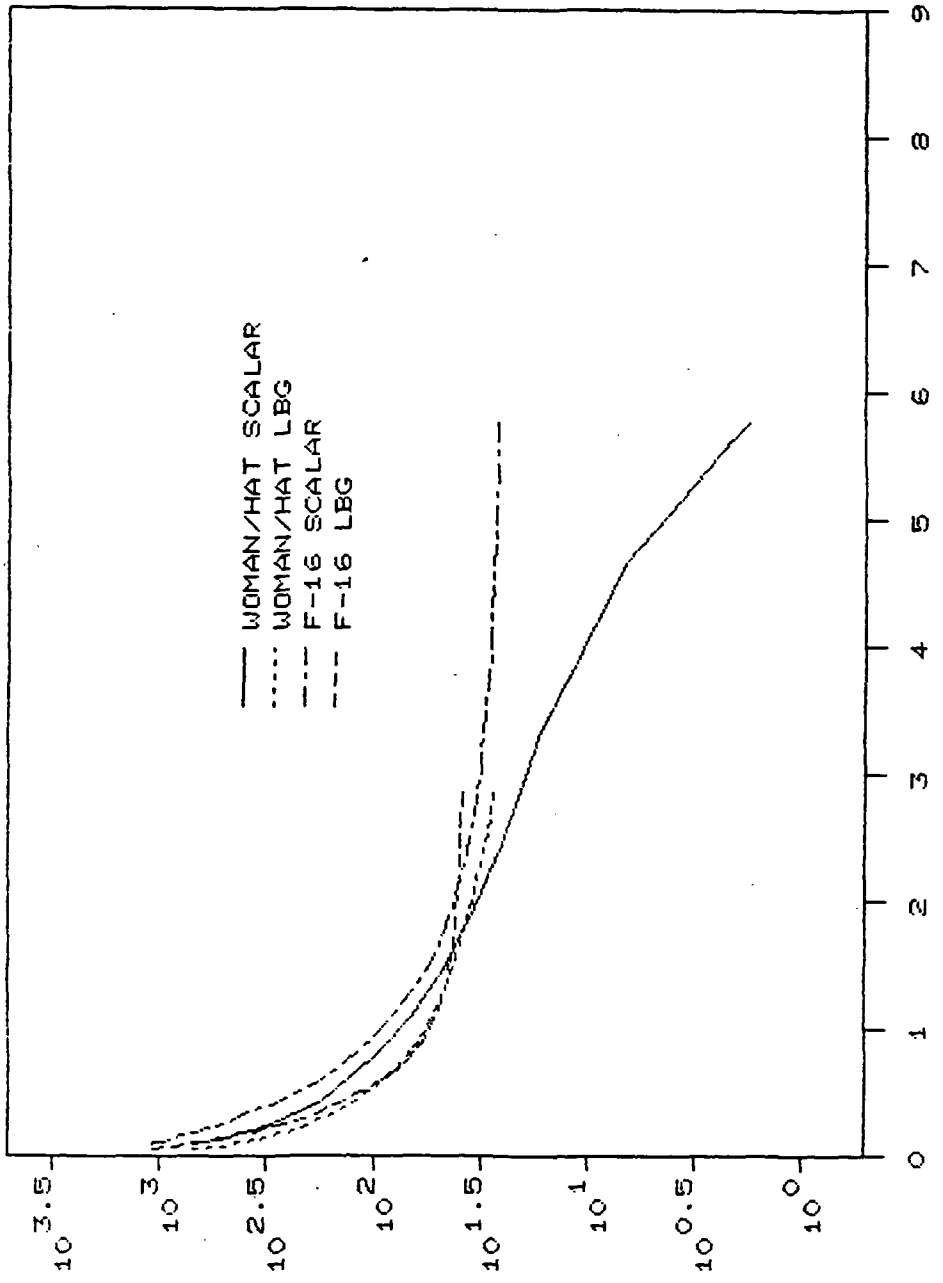


Figure 5.23 MBC distortion versus rate for RGB woman/hat and F-16.

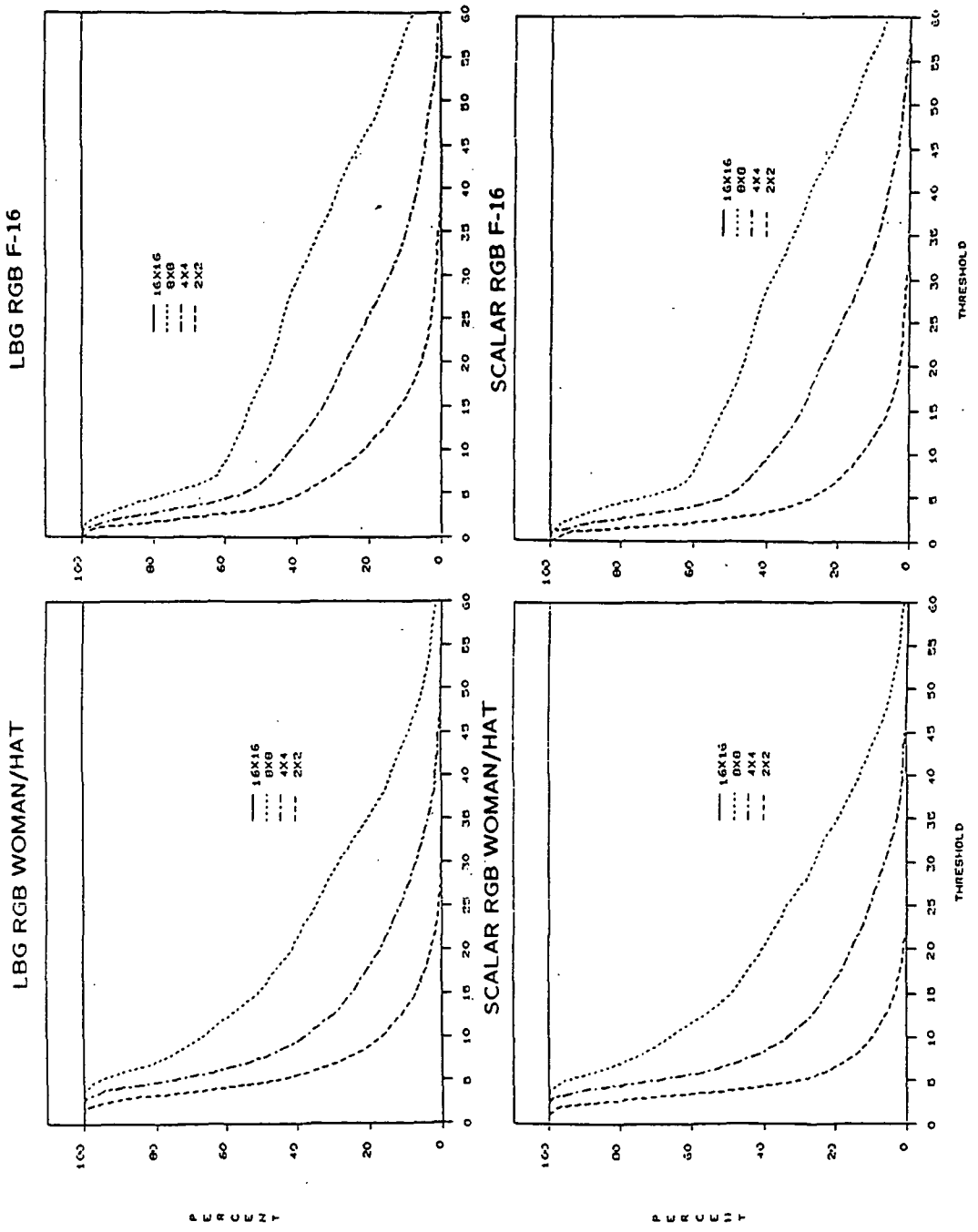


Figure 5.24 MBC/PT mixture fractions versus threshold for RGB woman/hat and F-16.

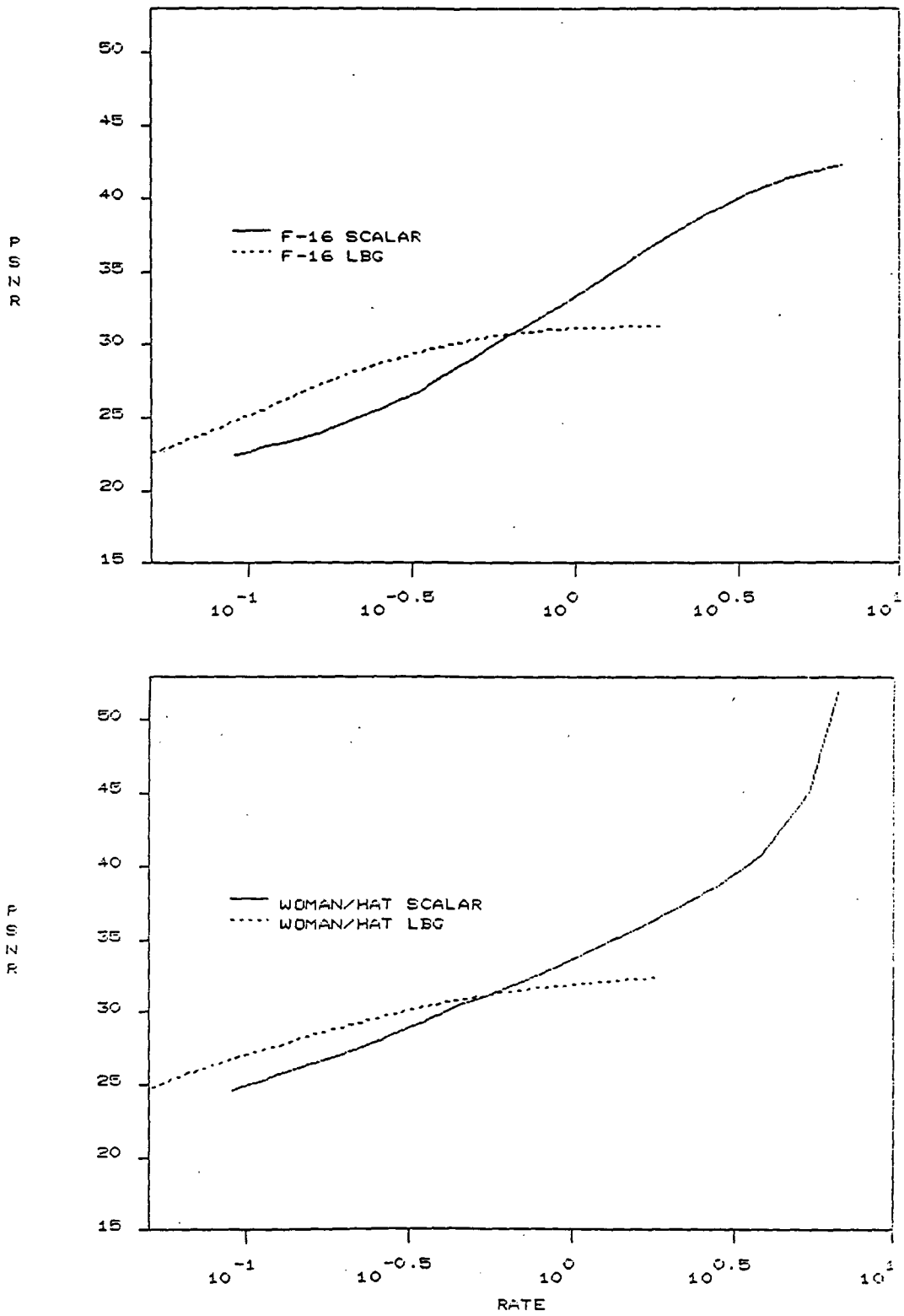


Figure 5.25 MBC/PT PSNR versus rate for RGB woman/hat and F-16.

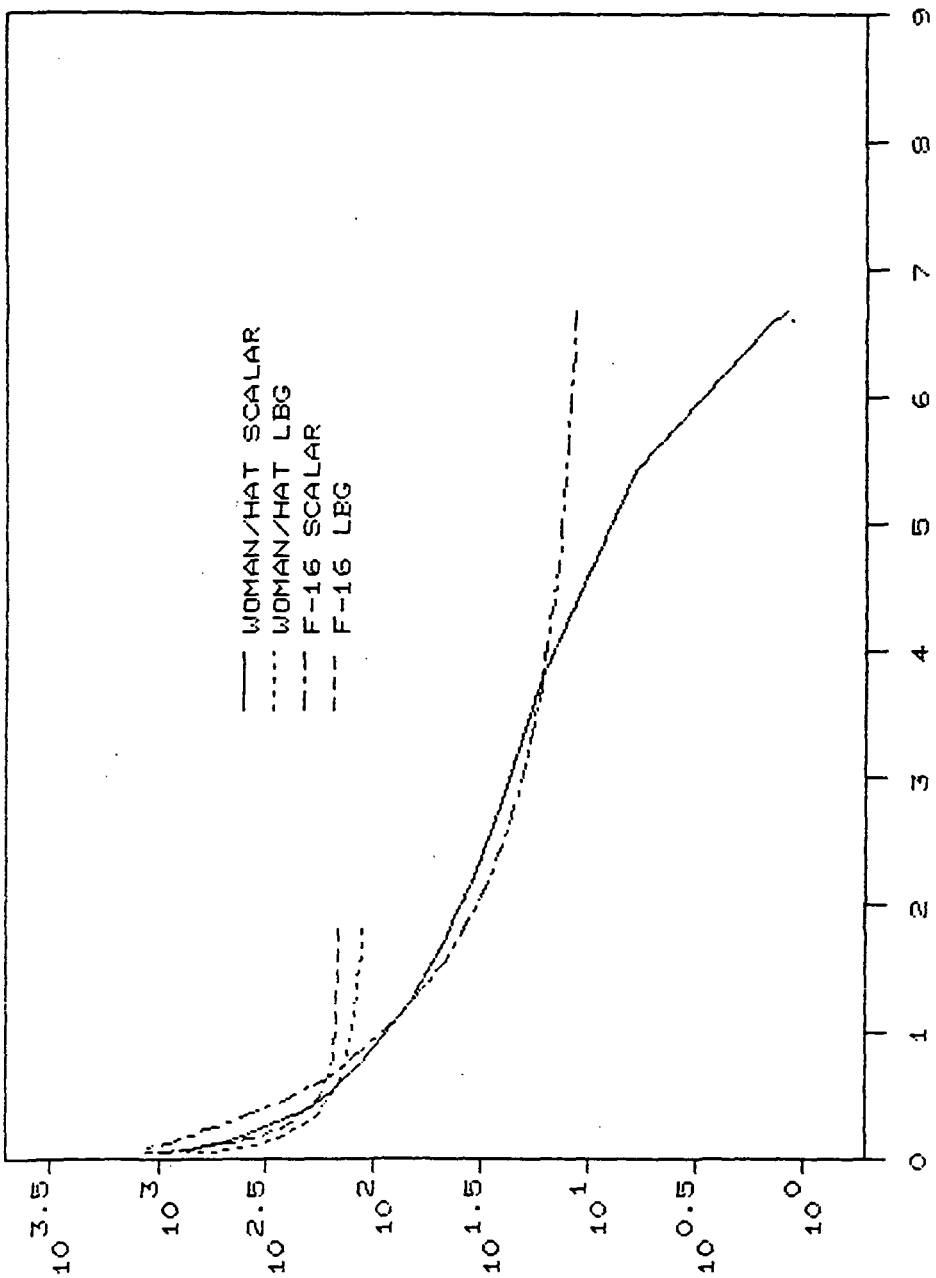


Figure 5.26 MBC/PT distortion versus rate for RGB woman/hat and F-16.

Figure 5.27 256x256 central portion of 512x512 BW woman/hat.

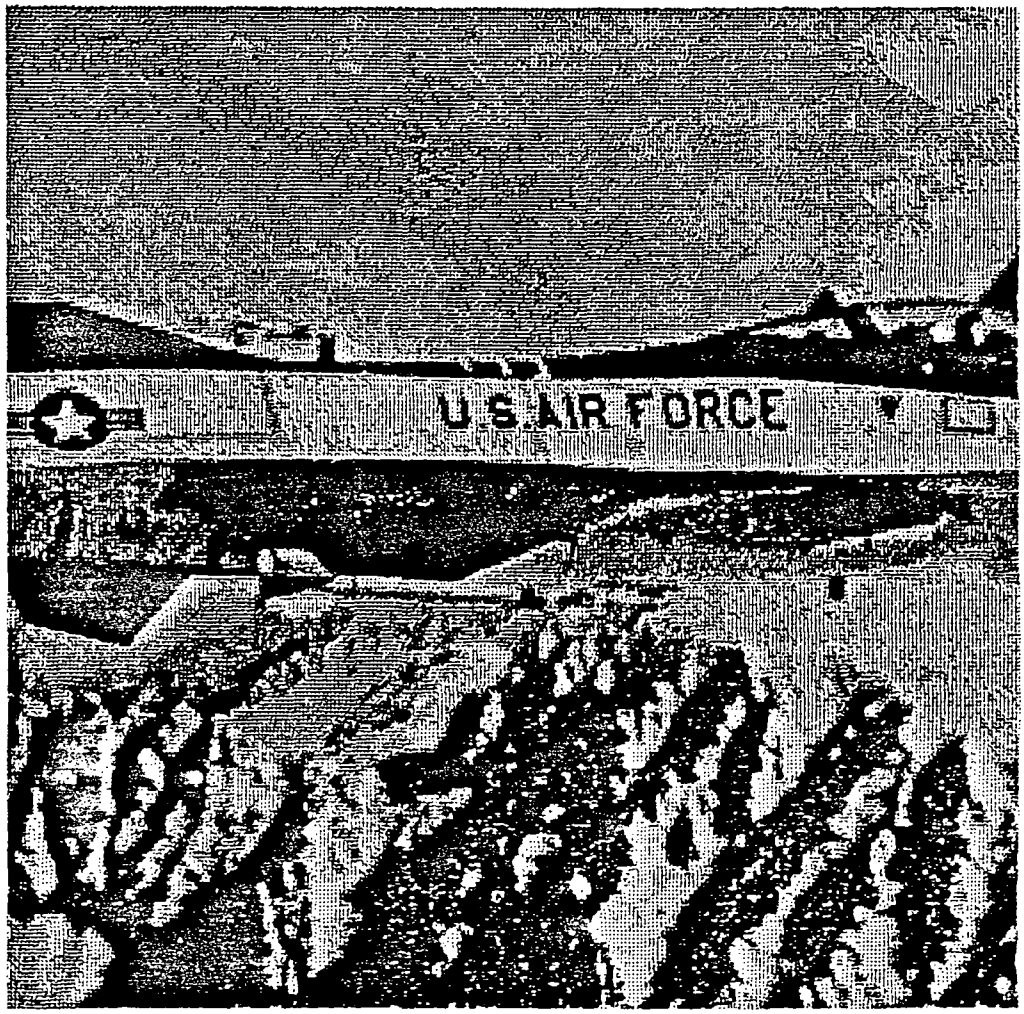


Figure 5.28 256x256 central portion of 512x512 BW F-16.

Figure 5.29 256x256 BW UCLA girl training image.

Figure 5.30 256x256 BW small face training image.

Figure 5.31 256x256 BW photo face training image.

Figure 5.32 256x256 central portion of MBC woman/hat.

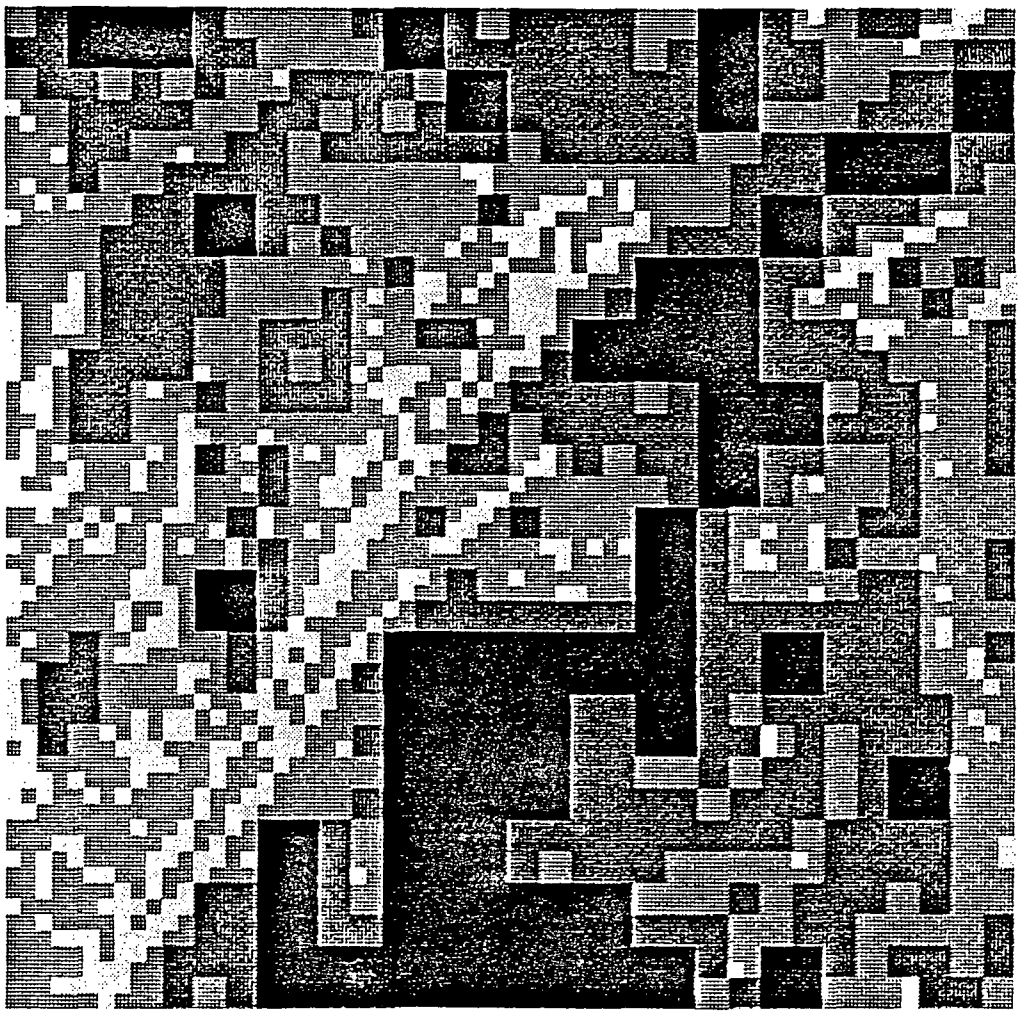


Figure 5.33 256x256 central portion MBC woman/hat block profile.

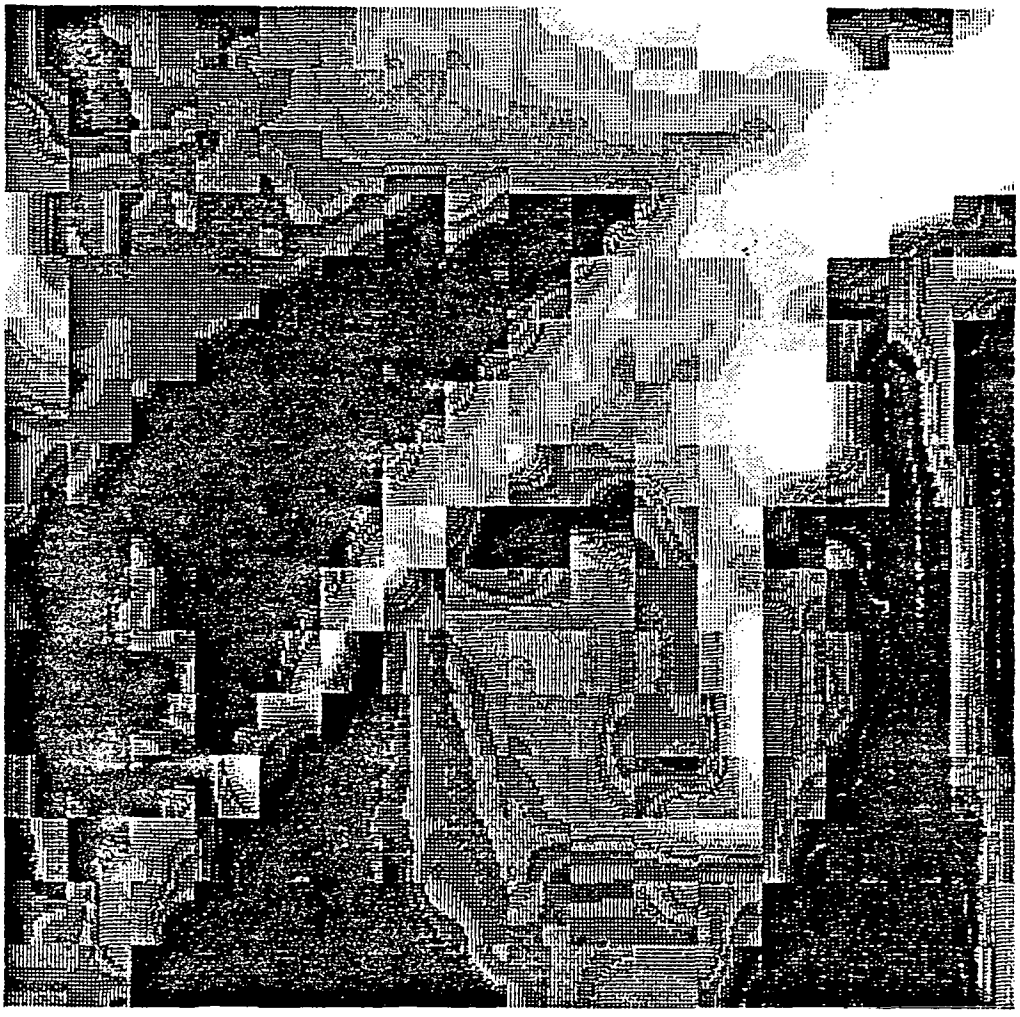


Figure 5.34 256x256 central portion of first pass MBC/PT woman/hat coded with 16x16 blocks.

Figure 5.35 256x256 central portion of MBC F-16.

Figure 5.36 256x256 central portion of MBC/BTC F-16.

Figure 5.37 256x256 central portion of MBC F-16 block profile.

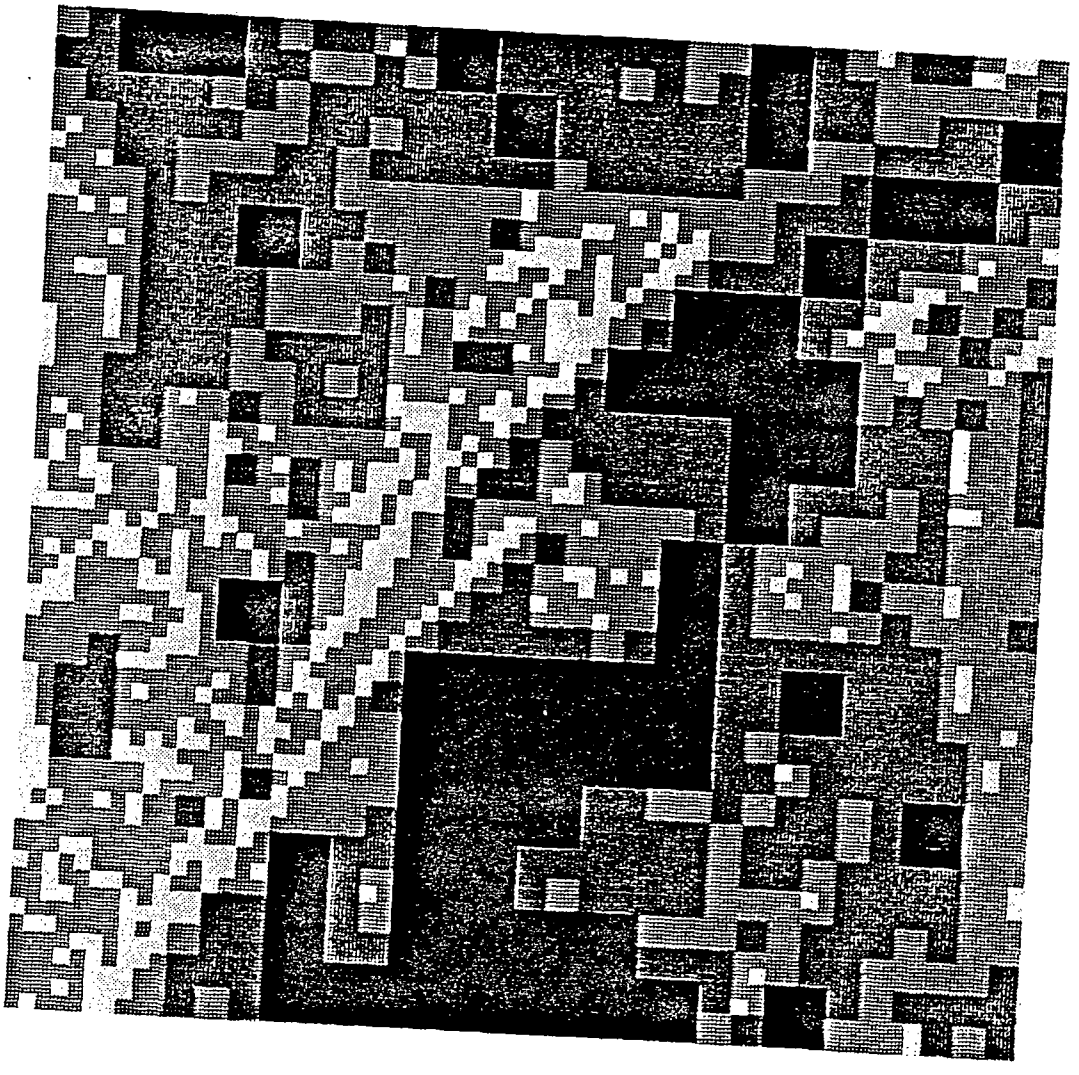


Figure 5.38 256x256 central portion of MBC/BTC F-16 block profile.

Figure 5.39 256x256 central portion of MBC F-16.

Figure 5.40 256x256 central portion of MBC/BTC F-16.

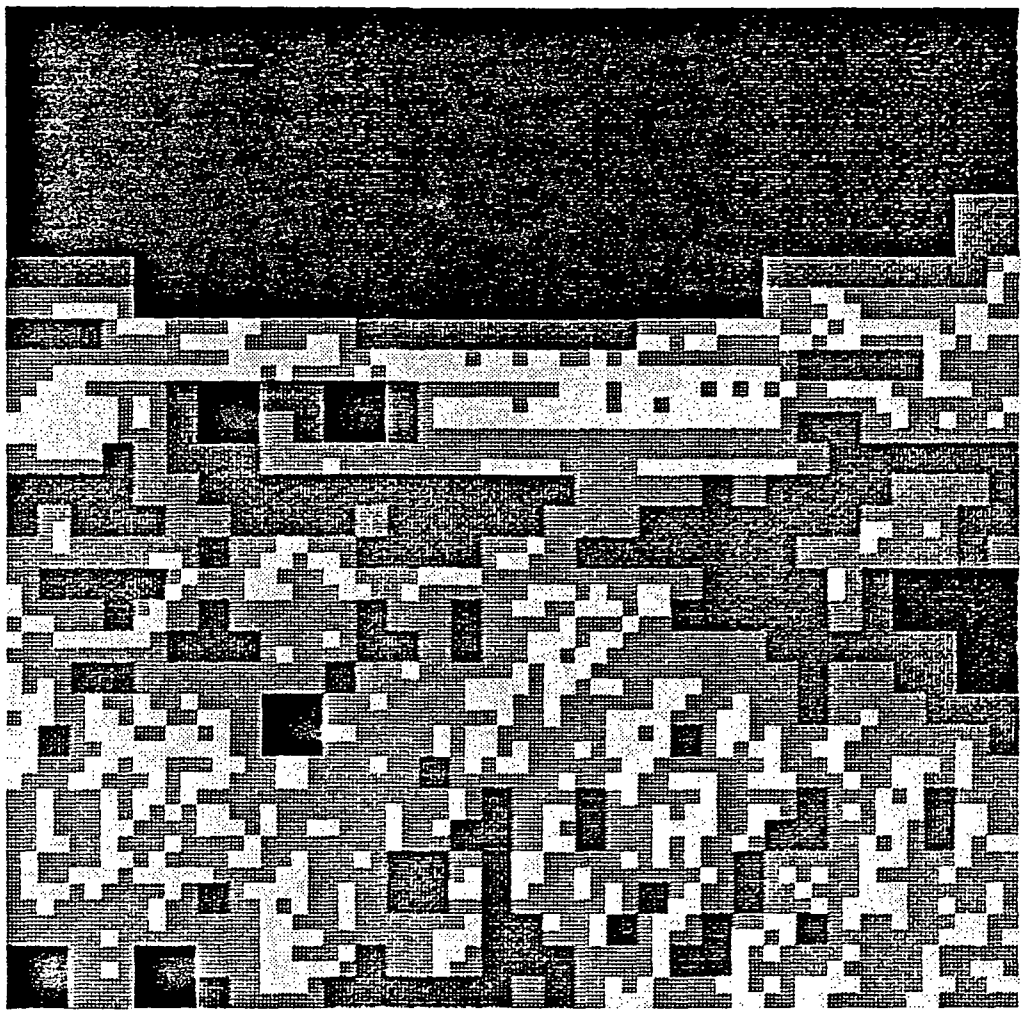


Figure 5.41 256x256 central portion of MBC F-16 block profile.

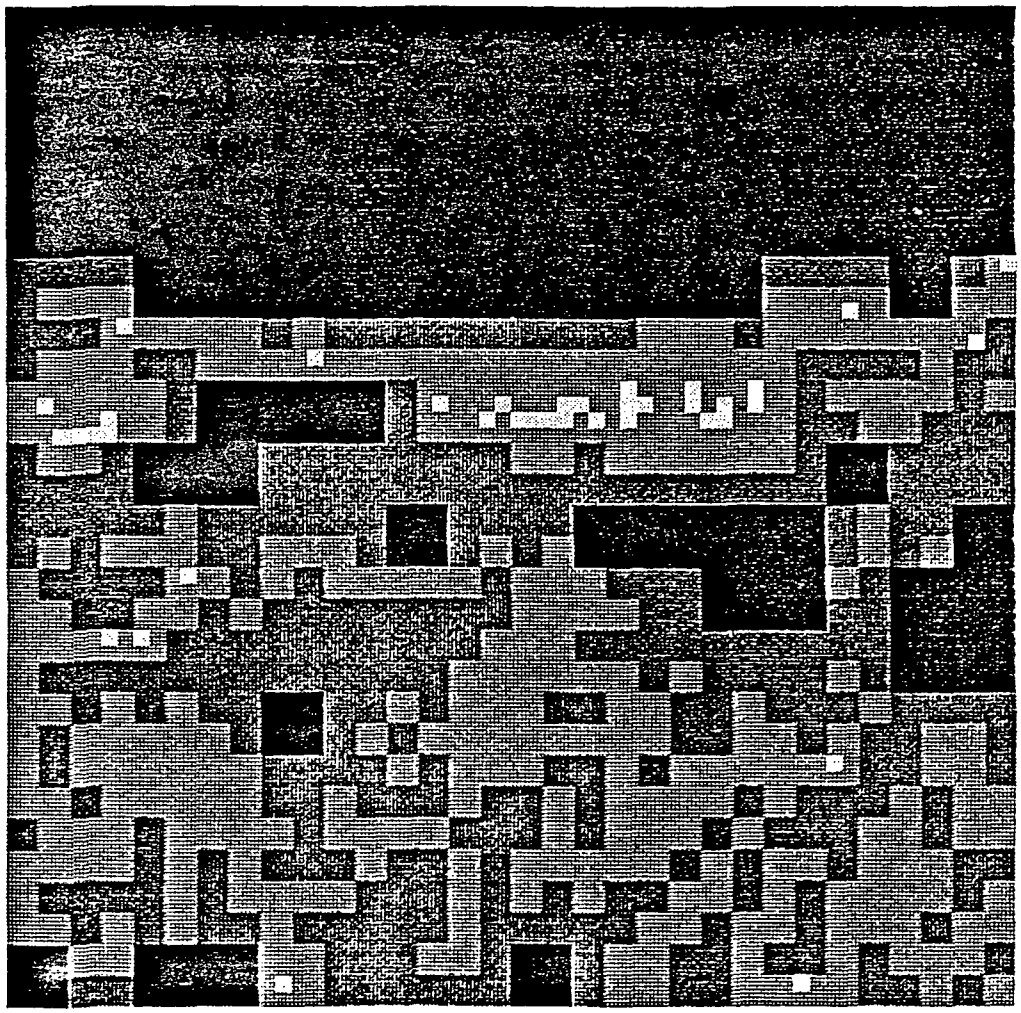


Figure 5.42 256x256 central portion of MBC/BTC F-16 block profile.

Table 5.1. Vector Quantizer Scalar Factors for BW Images

Blocksize	scale factor
16×16	60
8×8	35
4×4	20
2×2	10

Table 5.2. Vector Quantizer Scale Factors for RGB Images

Blocksize	scale factor
16×16	60
8×8	40
4×4	20
2×2	10

Table 5.3. Vector Quantizer Scale Factors for YIQ Images

Blocksize	scale factor
16×16	80
8×8	40
4×4	20
2×2	10

Chapter 6.

A Distortion-rate Function for Transform Coding

In this chapter, a distortion-rate function for transform source coders is developed. The assignment of the bit allocations for scalar quantized transform coefficients is the prime motivation for this material. Transform techniques can offer a significant coding advantage. This is because of their ability to redistribute the source signal energy in a relatively small number of easily identifiable coefficients (e.g., the low frequency coefficients). It is advantageous to code these high energy coefficients with more bits than the low energy coefficients. This is the basis for the approach taken for the bit assignment methods developed in this chapter.

To start with, the implementational problems that limit the use of the most commonly found bit assignment solution from the literature are discussed. Then methods are developed to overcome them. In the next chapter, the distortion-rate function developed is expanded to include a formulation that can be used for vector quantizers and MBC systems.

This presentation uses a distortion function introduced by Huang and Schultheiss [70] in 1963, that relies upon an observation of Max [71] from 1960. (Their function and this observation are discussed below.) Huang and Schultheiss originally used their distortion function for computing the bit allocations to select the scalar quantizer that used to code transform coefficients. Their method has been used with very little modification as recently as this year [72]. Their method uses assumptions that make the bit allocation algorithm easy to understand. But, these assumptions present some obvious implementation problems.

Their method allows for fractional and negative bit allocations for the quantized coefficients. The negatively assigned rate problem is especially troublesome when coding at very low rates. These problems were mentioned by Huang and Schultheiss, but they did not attempt to

resolve them. Instead, they used an iterative trial-and-error method to do the actual weighting selection. In 1979, Segall published some results that use the non-negative bit allocation constraint, but his work studies only the case for gaussian sources where it is possible to invert the dual optimization functional. (This functional results from the Lagrange multiplier. More will be said about this below.) A method developed by Shoham and Gersho [25] for computing optimal integer bit assignments for quantizers has appeared recently in the literature. Their method seems to require that rate be a monotonic function of the Lagrange multiplier that is associated with the average rate constraint.

Developing a more complete solution to these problems is the subject of this chapter. The negative bit allocation problem is overcome by adding a set of constraints to the distortion-rate function. Then, it is possible to obtain an implementable source coder to overcome the fractional bit allocation problem by using a linear mixture of two different scalar quantizers.

A bonus obtained from the reformulation is a more realistic set of upper and lower bounds for the distortion-rate function. These functions are expanded in the next chapter to include the use of vector quantization methods for the source coding of transform coefficients.

6.1 The distortion-rate problem

The goal of a source coder is to maximize the information that can be obtained about the source when looking at source coder output (reproducing alphabet). The distortion-rate bound predicts the least amount of distortion one can expect when coding with a predetermined maximum rate. Consider the following definitions in preparation for the formulation of the distortion-rate function.

Let there be a discrete source of n symbols with probabilities P_j which are to be coded with m reproducing symbols. The probability of the k-th reproducing symbol is a function of the source probabilities,

$$Q_k = \sum_{j=1}^{n} Q_{kj} P_j \tag{6-1}$$

where the Q_{kj} are the source coder transition probabilities. The probability of coding the j-th source symbol with the k-th reproducing symbol is Q_{kj}. Source coding distortion, d(Q), is a function of the transition probabilities and it is desirable to select the Q_{kj} to minimize its value. Letting ρ_{jk} be the distortion incurred when the j-th source symbol is coded with the k-th reproducing symbol, the total coding distortion is

$$d(Q) = \sum_{j=1}^{n} \sum_{k=1}^{m} \rho_{jk} Q_{kj} P_j \tag{6-2}$$

The average mutual information, I(Q), is a measure of the amount of information obtained about the source by looking at the reproducing alphabet. It is also a function of the transition probabilities,

$$I(Q) = \sum_{j=1}^{n} \sum_{k=1}^{m} Q_{kj} P_j \log_2 \frac{Q_{kj}}{Q_k} \tag{6-3}$$

This form of the mutual information assumes the reproducing alphabet is coded with binary symbols. This assumption is used in the sequel.

The distortion-rate problem is to select the transition probabilities to minimize the distortion incurred when coding for a given maximum mutual information or rate, R. More formally,

$$D(R) = \min_{R \epsilon A_R} d(Q) \tag{6-4}$$

where A_R is the set of admissible rates determined by the transition probabilities

$$A_R = \{Q_{kj} \mid I(Q) \leq R\} \tag{6-5}$$

Since the distortion-rate function, D(R), is one-to-one transformation [74], the distortion-rate problem will give the same solution for rate, given the distortion, as is indicated in (4)

$$R(D) = \min_{D \epsilon A_D} I(Q) \tag{6-6}$$

where A_D is the admissible distortions determined by the transition probabilities

$$A_D = \{Q_{kj} \mid d(Q) \leq D\} \tag{6-7}$$

The distortion-rate function and the rate-distortion function can be used interchangeably.

To find the optimal transition probabilities for the D(R) function one must know:

- the number of source symbols and their probabilities of occurrence, P_j, and
- the coding distortions, ρ_{jk}.

The problem seems simple on the surface, but there are several caveats concerning the uses of these functions when studying actual source coders that must be noted:

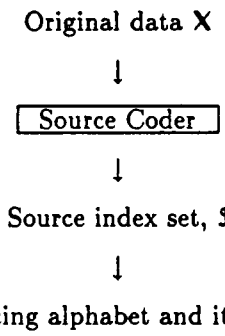
- The sources considered here are continuous, and not discrete as indicated above. Allowances must be made to include this situation.
- The coding distortions are, of course, a function of the source coding scheme. Here they must represent quantization losses.
- Typically, the transition probabilities are not constrained in any way (other than conforming to the rules of probability assignment) and they are used to develop bounds for the performance of generalized source coders. The results obtained say nothing about the complexity of implementing a coder to meet the predicted distortion levels. Since the goal is to study the performance of a specific type of coder its particular functional needs must be understood and modified accordingly.

The work of Huang and Schultheiss and of Max leads one to a particularly useful model for distortion when using scalar quantization for a known continuous source. They show distortion is a function of the coefficient source variance (σ_i^2), the number of quantizer coding levels (2^{-2B_i} where B_i is the i-th coefficient rate), and its quantizer performance factor (ϵ_i^2). For the work done here, the source is a set of transform coefficients whose pdf is assumed laplacian. (This is consistent with the characteristics of the data found in Chapter 5.) More is said about the characteristics of the laplacian performance factor later. Using these definitions, the distortion incurred when coding the i-th transform coefficient when using B_i bits is

$$d_i = \epsilon_i^2 \sigma_i^2 2^{-2B_i} \tag{6-8}$$

The transition probabilities, as used in the distortion-rate bound, are assigned without concern for how they are to be used within a given design. They represent lower bounds for any realizable coding system, and may not be attainable in any particular application. When designing a usable source coder, several practical problems arise when using this approach. These problems are demonstrated below.

In the following discussion consider the following diagram.



For a given source coder distortion level, the source coder mutual information (3) is the lower bound for the source coder rate, R, no matter how the source index symbols are coded. This fact is the result of the distortion-rate function, as is indicated by (4) and (5),

$$I(Q) \leq R \tag{6-9}$$

Now, let C be the map which chooses the actual symbols used to represent the source index set. The output of map C is called the source coder reproducing alphabet. The entropy of these symbols, $H_c(\mathcal{J})$, is the lower bound for the actual rate, R_c, obtained. Since R is the lower bound for any C mapping, $R \leq H_c(\mathcal{J})$ and

$$I(Q) \leq R \leq H_c(\mathcal{J}) \leq R_c \tag{6-10}$$

But, the source rate R used in the distortion-rate function D(R), is not necessarily attained when the realities of the C mappings are considered. To see how this is possible, consider the details of how the rate R_c is assigned.

$H_C(\mathfrak{I})$ is a function of how C maps the elements of the quantizer index into a reproducing alphabet. For example, define a reproducing alphabet by grouping the quantizer index symbols into k-vectors, or k-blocks,

$$v_j = (i_{j1}, i_{j2}, \ldots, i_{jk}) \varepsilon Z^k \tag{6-11}$$

The entropy of these blocked symbols is

$$H_k(\mathfrak{I}) = -\sum_{j=1}^{m^k} p(v_j) \log_2 p(v_j) \tag{6-12}$$

$H_k(\mathfrak{I})$ is known to have the following property [74]

$$H_k(\mathfrak{I}) \leq \ldots \leq H_2(\mathfrak{I}) \leq H_1(\mathfrak{I}) \tag{6-13}$$

The entropy of symbols grouped in blocks decreases monotonically with the length of the block. Letting k be the length of the vectors used to drive the actual symbol map, C. Since the R is determined by the distortion-rate function it is the lower bound (10) for any length block coder

$$R \leq H_C(\mathfrak{I}) \equiv H_k(\mathfrak{I}) \tag{6-14}$$

The symbol rate may never reach R, no matter how long its vectors become. The complexity of the symbol map used is determined by the length of these vectors, and if they are too long the transmitter design can become overwhelming.

Using rate-distortion theory to solve problems that set bounds upon the set of all possible coders, like the one above, can be very useful. But, rate-distortion theory leaves something to be desired when studying a particular class of source coders were the transition probabilities are of little or no help. This is true because sometimes the mutual information is of marginal use in the formulation of the distortion-rate solution. It would be helpful if the mutual information could be replaced with another rate formulation that is of more use.

For the work of this dissertation, it is very helpful if the transition probabilities are replaced with the transform coefficient rates, $\{B_i\}$. For the scalar quantized systems of this chapter, it is of interest to study the distortion-rate problem where each coefficient is coded individually. The distortion-rate function can be optimized over this set. The admissible set of

all possible coefficient rate allocations given an average coefficient coding rate, B, is

$$A_B = \{B_i \mid \sum_i B_i = nB,\ B_i \geq 0\} \tag{15}$$

where n is the number of coefficients. Vector quantization of transform coefficients is studied in the next chapter. There the B_i represent the coding rate for the i-th block of coefficients.

With the above considerations in mind, the distortion-rate problem is ready to be presented and solved for the different variations that are important to this dissertation.

6.2 The transform coefficient distortion-rate problem

The problem is to minimize the total quantization distortion obtained when coding a set of n transform coefficients given an average rate constraint. The distortion-rate problem can be stated as a sum of distortion terms of the form (8), and it is to be solved with two rate constraints (15):

$$\min_{\{B_i\}} D = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2B_i} \tag{6-16}$$

subject to

$$\sum_{i=1}^{n} B_i = nB \text{ and} \tag{6-17a}$$

$$B_i \geq 0\ \forall i \tag{6-17b}$$

where B is the average coefficient rate. The B_i, σ_i^2, and ϵ_i^2 are, respectively, the i-th coefficient rate, variance, and quantizer performance factor. The number of quantization levels for the i-th coefficient is 2^{B_i}. The summation constraint (17a) sets the total rate used to code the n transform coefficients, and the non-negativity constraints (17b) guarantee none of the coefficient rates are allowed to be less than zero. It is assumed the solution space defined by (17a) and (17b) is not empty.

The problem as formulated by Huang and Schultheiss in 1963, does not include the non-negativity constraints. This facilitates a simpler solution because only the single summation constraint needs to be considered, instead of the n+1 constraints of the system indicated above. Consider what this means.

If coefficient rates are allowed to go negative not only is the system unrealizable but, (16) shows the coding of these coefficients returns a larger distortion summand than if no bits were used to code them

$$\epsilon_i^2\sigma_i^2 2^{-2B_i} > \epsilon_i^2\sigma_i^2 \geq \sigma_i^2 \text{ when } B_i<0$$

(For scalar quantizers, the $\epsilon_i^2 \geq 1$ as is shown later.) This "greater-than-no-coding" loss is balanced by allocating larger rates to other coefficients that have positive allocations. Consider the simple example that is presented in Appendix 2.

When performing scalar quantization it is not possible to code a coefficient with less than zero bits. The usual practice is to round the negative allocations to zero and redistribute the total number of bits among the other coefficients in near proportion to their associated rate weightings. This amounts to reworking the mathematically predicted optimal coefficient weighting problem in an ad hoc fashion which, in effect destroys the optimality of the solution.

The inability to correctly handle negative bit allocations is why it is necessary to include the non-negativity constraints in the distortion-rate function. But, at least to start, only the summation constraint is used. This is done to demonstrate the solution most commonly found in the literature (e.g., [72]). Once the sum-constrained solution is shown, we go back and add the non-negativity constraints and reformulate the solution to include them. These added constraints do not complicate the solution excessively when it is approached with care, and it offers a more realistic distortion-rate function because it is more feasible from an implementation point of view. Since this solution is more realistically constrained than the Huang and Schultheiss solution, these distortion-rate bounds are of greater value.

6.3 The simplest rate solutions

The sum-constrained solution of (16) and (17) can be found by setting the rate derivatives of the Lagrangian to zero,

$$\frac{\partial}{\partial B_i}\left\{D - \lambda\left(nB - \sum_{j=1}^{n} B_j\right)\right\} = 0 \quad \forall i \tag{6-18}$$

and solving for the B_i. Thus,

$$\frac{\partial D}{\partial B_i} + \lambda = 0 \quad \forall i \tag{6-19}$$

and since

$$\frac{\partial D}{\partial B_i} = (-2\ln 2)\epsilon_i^2 \sigma_i^2 2^{-2B_i} + \frac{\partial \epsilon_i^2}{\partial B_i}\sigma_i^2 2^{-2B_i} \quad \forall i \tag{6-20}$$

the optimal rates are assigned through the solution of these differential equations:

$$\sigma_i^2 2^{-2B_i}\left\{(-2\ln 2)\epsilon_i^2 + \frac{\partial \epsilon_i^2}{\partial B_i}\right\} + \lambda = 0 \quad \forall i \tag{6-21}$$

The common assumption is to set the partial derivatives of ϵ_i^2 to zero [70]. This means the performance factors are assumed to be constant. The performance factors can be found by fitting a curve of the form $\epsilon^2 2^{-2B}$ to the quantization data of the particular pdf desired. This fit matches the data very well for large rates. The Gaussian, Laplacian and gamma scalar optimal (nonuniform) quantizer performance factors are shown in Figure 6.1. These curves are constructed using the distortion data presented in [71,73,75]. Notice that they all start at one and approach an asymptotic limit as the rate goes toward infinity. The asymptotic limit is used in the literature to estimate the performance of these quantizers. Notice that the actual distortion is less than this limit. Functions that use this assumption give a solution that is an upper bound to the performance that is actually obtained. Rate-distortion theory states that the best performance factor is 1, and that it is obtainable only for the gaussian pdf. This bounds is obtained using the Shannon lower bound, see Berger [74] for details.

The zero-derivative assumption is sometimes justified (e.g., [46]) since the performance factors of the common optimal scalar quantizers vary more slowly than the associated 2^{-2B_i} term of the distortion equation. For large data rates this assumption is justified but, for low rates it does not always hold. If the assumption is used and all of the coefficient pdf's are the same, then all of ϵ_i^2 are equal. As a result they drop out of the final solution, as is shown next.

Using the $\frac{\partial \epsilon_i^2}{\partial B_i}$=0 assumption, and setting ϵ_i^2 to, say, ϵ^2 for the case where all of the coefficients have the same pdf, (21) becomes

$$(-2\ln 2)\epsilon^2\sigma_i^2 2^{-2B_i} + \lambda = 0 \quad \forall i \tag{6-22}$$

These equations can be solved for the B_i,

$$B_i = \tfrac{1}{2}\log_2(2\ln 2) + \tfrac{1}{2}\log_2\epsilon^2\sigma_i^2 - \tfrac{1}{2}\log_2\lambda \quad \forall i \tag{6-23}$$

Using the summation constraint, λ can be found

$$nB = \tfrac{n}{2}\log_2(2\ln 2) + \sum_{j=1}^{n}\tfrac{1}{2}\log_2\epsilon^2\sigma_j^2 - \tfrac{n}{2}\log_2\lambda \tag{6-24}$$

$$\tfrac{1}{2}\log_2\lambda = \tfrac{1}{2}\log_2(2\ln 2) + \tfrac{1}{n}\sum_{j=1}^{n}\tfrac{1}{2}\log_2\epsilon^2\sigma_j^2 - B \tag{6-25}$$

Substitute (25) into (23) to find the i-th coefficient rate

$$B_i = B + \tfrac{1}{2}\log_2\sigma_i^2 - \tfrac{1}{n}\sum_{j=1}^{n}\tfrac{1}{2}\log_2\sigma_j^2 \tag{6-26}$$

$$= B + \tfrac{1}{2}\log_2\frac{\sigma_i^2}{\left(\prod_{j=1}^{n}\sigma_j^2\right)^{1/n}} \tag{6-27}$$

The bit rates are allocated using the ratio of the coefficient variance to the geometric mean of all the source coefficient variances. Notice the coefficient rates are linearly proportional to the average rate. A set of curves representing the coefficient rates of (27) using four coefficients whose variances are 1, 4, 16, and 64, is shown in Figure 6.2.

If the ϵ_i^2 are still assumed to be constant, but different for each coefficient as would be the case when the individual pdf's vary with i, then the coefficient rates are

$$B_i = B + \tfrac{1}{2}\log_2\frac{\epsilon_i^2\sigma_i^2}{\left(\prod_{j=1}^{n}\epsilon_j^2\sigma_j^2\right)^{1/n}} \tag{6-28}$$

In reality, the value of various performance factors for the different types of scalar quantizers varies with rate, especially for low rates, and their derivatives need to be considered in the solution of (17). For scalar quantizers, the value of $\epsilon_i^2(R)$ is computable only for the rates $R=\log_2 L$, where L is an integer representing the number of code book elements. The value of the rate-dependent performance factor, $\epsilon_i^2(R)$, for rates not in this set can be found using linear

interpolation. For simplicity, consider the case when only scalar quantizers of integer coding rate are considered (that is, L = 0, 2, 4, 8, . . ., for the coding rates of 0, 1, 2, 3, . . . bits/sample). To obtain a coding rate that is not integral, it is possible to use a α-mixture of two integer quantizers. For example, if B is an integer, then for rates between B and B+1 the distortion performance of such an α-mixture quantizer is

$$D(B+\alpha) = (1-\alpha)D(B) + \alpha D(B+1) \quad 0 \leq \alpha < 1 \tag{6-29a}$$

where $D(R)$ is the expected distortion for the quantizer at rate R. This equation indicates the quantizer distortion for nonintegral rates is a mixtures of the two end-point quantizers. A quantizer that codes at the rate of, say, 1.2 bits can be obtained by using a 1-bit quantizer 80 percent of the time, and a 2-bit quantizer 20 percent of the time. Notice that the distortion-rate function described by (29a) is a straight-line function between its integer end points.

The quantizer mixture of (29a) is easy to implement using the quantizer data that is available in the literature (e.g., [71,73,75]). The low-rate performance factors for these quantizers are shown in Figure 6.1. It is possible to derive a model for the performance factor predicted by (29a) using the distortion model

$$D(R)=\epsilon_i^2(R)2^{-2R} \tag{6-29b}$$

Equation (29a) becomes

$$\epsilon_i^2(B+\alpha)2^{-2(B+\alpha)} = (1-\alpha)\epsilon_i^2(B)2^{-2B} + \alpha\epsilon_i^2(B+1)2^{-2(B+1)} \tag{6-29c}$$

and the associated performance factor is

$$\epsilon_i^2(B+\alpha) = (1-\alpha)\epsilon_i^2(B)2^{2\alpha} + \alpha\epsilon_i^2(B+1)2^{-2(1-\alpha)} \tag{6-29d}$$

For high coding rates the performance factors of optimal scalar quantizers approach constant values. (Let ϵ_*^2 be the high-rate performance factor value: then gaussian ϵ_*^2=1.00, Laplacian ϵ_*^2=4.50 and gamma ϵ_*^2=5.68, Jayant and Noll [1, Table 4.8],). This can be used to develop a high-rate performance factor

$$\epsilon_{i\infty}^2(B+\alpha) = \epsilon_i^2(B)\left((1-\alpha)2^{2\alpha} + \alpha 2^{-2(1-\alpha)}\right) \text{ as } B\rightarrow\infty \tag{6-29e}$$

These facts will be used later to compute a distortion-rate function for transform source coders that are scalar quantized.

Since solutions using rate-variable performance factors are not known for negative rates a solution for the distortion-rate function using this feature must wait until the non-negativity constraints are applied. This is the subject of the next section.

6.4 The non-negative rate solutions

Before finding the distortion-rate solution using the non-negativity constraints, consider what happens to the individual coefficient rates (28) as the average rate varies. Since (28) is linearly proportional to rate, it can be simplified to

$$B_i = B + \alpha_i \tag{6-30}$$

where

$$\alpha_i = \frac{1}{2}\log_2 \frac{\epsilon_i^2 \sigma_i^2}{\left(\prod_{j=1}^{n} \epsilon_j^2 \sigma_j^2\right)^{1/n}} \tag{6-31}$$

Here α_i is a function of the i-th coefficient variance, but not the rate. Notice that the i-th coefficient rate has unit slope, and is not positive for all $B \leq \alpha_i$. For large rates greater than $-\alpha_i$ all of the coefficient rates are positive. This fact is capitalized upon in the sequel. To simplify the following equations, let it be assumed, without loss of generality, that the coefficients are arranged in descending order of variance,

$$\sigma_i^2 \leq \sigma_j^2 \qquad \forall i > j \tag{6-32a}$$

Since the α_i are monotonically decreasing functions of the variance, they are arranged in ascending order

$$\alpha_i \geq \alpha_j \qquad \forall i > j \tag{6-32b}$$

Now, consider a few definitions concerning the solution space defined by the constraints (17a) and (17b). For a given rate, the set of points satisfying both (17a) and (17b) represents the set of all possible solution points over which the minimization of (16) can be taken. Any

point satisfying (17a) and (17b) is called a *feasible* point [76]. If, for some feasible point, one of inequality constraints of (17b) is exactly satisfied (e.g., $\exists$i such that B_i=0), then the associated point lies on the boundary of the set of feasible points. This exactly-satisfied constraint is said to be *active*. If any constraint is not active (e.g., B_i>0), then the associated feasible point does not lie on that particular boundary. Such a constraint is said to be *inactive*.

When all of the non-negativity inequalities are inactive, all of the coefficient rates are strictly positive, and the coefficient rates of (31) are trivially equal to those of (28). This is true for all $B \geq -\alpha_n$. For consistency, let the definition of α_i of (31) be extended to include notation that explicitly indicates the number of inequality constraints that are active. Let $\alpha_i^{(j)}$ be the value of α_i when j inequality constraints are active. Then the α_i of (31) becomes

$$\alpha_i^{(n)} = \frac{1}{2}\log_2 \frac{\epsilon_i^2 \sigma_i^2}{\left(\prod_{j=1}^{n} \epsilon_j^2 \sigma_j^2 \right)^{1/n}} \tag{6-33}$$

As the rate is gradually reduced to values less than $-\alpha_n^{(n)}$, the inequality constraints become active starting with the n-th constraint. As the rate becomes smaller each successive constraint becomes active (e.g., n−1, n−2, . . .) until the rate reaches zero where all of the constraints are active. These facts allows the non-negatively constrained rates to be found within each of the different activity intervals.

Consider the case where only the n-th inequality constraint is active. In this case, B_n=0 for some coding rate $B \leq -\alpha_n^{(n)}$ and all of the other coefficient rates are greater than zero. For these coding rates, taking the more general case where the performance factors are constant but different for each coefficient, equations (24) and (25) become

$$nB = \frac{n-1}{2}\log_2(2\ln 2) + \sum_{j=1}^{n-1} \frac{1}{2}\log_2 \epsilon_j^2 \sigma_j^2 - \frac{n-1}{2}\log_2 \lambda \tag{6-34}$$

and

$$\frac{1}{2}\log_2 \lambda = \frac{1}{2}\log_2(2\ln 2) + \frac{1}{n}\sum_{j=1}^{n} \frac{1}{2}\log_2 \epsilon^2 \sigma_j^2 - \frac{n}{n-1}B \tag{6-35}$$

so the remaining n−1 nonzero coefficient rates are

$$B_i = \frac{n}{n-1}B + \frac{1}{2}\log_2 \frac{\epsilon_i^2\sigma_i^2}{\left(\prod_{j=1}^{n-1}\epsilon_j^2\sigma_j^2\right)^{1/(n-1)}} \tag{6-36}$$

To simplify the notation, let this be rewritten using the notation of (33),

$$B_i = \frac{n}{n-1}B + \alpha_i^{(n-1)} \quad \forall i<n \tag{6-37}$$

$$B_n = 0$$

Notice that these solutions are straight-line segments of slope n/(n−1). They are valid in the interval from where the n-th inequality constraint became active to where the (n−1)-th inequality constraint becomes active. The (n−1)-th inequality constraint becomes active when B_{n-1} of (37) goes to zero,

$$B_{n-1} = \frac{n}{n-1}B + \alpha_{n-1}^{(n-1)} = 0 \tag{6-38}$$

Solve (38) for the average rate for which this is true

$$B = -\frac{n-1}{n}\alpha_{n-1}^{(n-1)} \quad \forall i<n \tag{6-39}$$

Therefore, the (n-1)-th activity interval is lower bounded by (39)

$$-\frac{n-1}{n}\alpha_{n-1}^{(n-1)} \le B < -\alpha_n^{(n)} \tag{6-40}$$

Now, for the case where only two inequality constraints are active, those for n and n−1, the coefficient rates become

$$B_i = \frac{n}{n-2}B + \frac{1}{2}\log_2 \frac{\epsilon_i^2\sigma_i^2}{\left(\prod_{j=1}^{n-2}\epsilon_j^2\sigma_j^2\right)^{1/(n-2)}} = \frac{n}{n-2}B + \alpha_i^{(n-2)} \quad \forall i<n-1 \tag{6-41}$$

$$B_i = 0 \text{ for } i = n, n-1$$

Similar to the case for (n−1)-th activity interval, the (n−2)-th activity interval end point can be found

$$-\left(\frac{n-2}{n}\right)\alpha_{n-2}^{(n-2)} \le B < -\left(\frac{n-1}{n}\right)\alpha_{n-1}^{(n-1)} \tag{6-42}$$

Within this interval the nonzero coefficient rates are straight line segments with slope n/(n−2).

By proceeding into each new activity interval the coefficient rates are all found in a similar fashion. When j inequality constraints are active, the i-th coefficient rates are

$$B_i = \frac{n}{n-j}B + \alpha_i^{(n-j)} \quad \forall i \leq n-j \qquad (6\text{-}43)$$

$$B_i = 0 \qquad \forall i > n-j$$

and these rates are valid over the interval

$$-\left(\frac{n-j}{n}\right)\alpha_{n-j}^{(n-j)} \leq B < -\left(\frac{n-j+1}{n}\right)\alpha_{n-j+1}^{(n-j+1)} \qquad (6\text{-}44)$$

The activity of the inequality constraints do not change within this interval. There are n+1 such intervals and they cover the entire set of all possible rates, $B \geq 0$.

Notice the slope of the coefficient rate increases as each inequality constraint becomes active. When all of the constraints are active, the coefficient slope are equal to n. This fact is illustrated in Figure 6.3, using the same four coefficient system of Figure 6.2. Before moving to the next subject, where the performance factors are allowed to vary with rate, the character of the distortion-rate function is explored for the rate solutions presented in this and the last section of the chapter.

6.5 Distortion function for constant performance factors

Consider the case for the distortion-rate function that is only sum constrained and the rates can be assigned negative values (there are no non-negativity constraints). As was shown above, the coefficient rates are assigned by equation (30)

$$B_i = B + \alpha_i \qquad (6\text{-}45)$$

so the distortion defined by (16) can be expressed in a particularly simple expression

$$D = \sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2(B+\alpha_i)} = \left(\sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i}\right) 2^{-2B} = \Theta 2^{-2B} \qquad (6\text{-}46)$$

The distortion is an exponential decay starting at Θ for the coding rate B=0.

The distortion function of (46) has some interesting properties. Consider the four coefficient example system whose rates are depicted in Figure 6.2. The distortion of each of the four distortion summands and the total distortion of (46) for this system are shown in Figure 6.3. Notice that all of these curves are straight lines on a semilog plot whose slope is −2. Define a

set of individual distortion functions, one for each coefficient

$$D = \sum_{i=1}^{n} D_i = \Theta 2^{-2B} \tag{6-47}$$

The semilog function of (47) is straight

$$\log_2 D = \log_2 \Theta - 2B \tag{6-48}$$

and so are those of each of the distortion summands

$$D_i = \left(\epsilon_i^2 \sigma_i^2 2^{-2\alpha_i}\right) 2^{-2B} = \Theta_i 2^{-2B} \tag{6-49}$$

$$\log_2 D_i = \log_2 \Theta_i - 2B \tag{6-50}$$

Now, consider the case where the non-negativity constraints are included. As they start to become active the distortion function of (47) must be modified to include the coefficient rates that become fixed at zero. For the case where none of these constraints are active ($B \geq -\alpha_n$), the distortion function of (47) is correct. For consistency with what is to follow, let (47) be redefined to include notation to show the number of active constraints

$$D^{(n)} = \left(\sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i^{(n)}}\right) 2^{-2B} = \Theta^{(n)} 2^{-2B} \tag{6-51}$$

The distortion in the rate interval $B \geq \alpha_n^{(n)}$, where none of the inequality constraints are active, is trivially equal to (51).

For the case where j non-negativity constraints are active, it will be shown that the distortion function has similar form to (51), within each interval of constant activity (44). Consider the case for which only the n-th inequality constraint is active. When only the n-th inequality constraint is active, the coefficient rates are defined by (37) becomes

$$B_i = \frac{n}{n-1} B + \alpha_i^{(n-1)} \quad \forall i < n \tag{6-52}$$

$$B_n = 0$$

and the distortion, as defined by (16), is

$$D^{(n-1)} = \sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2B_i} + \epsilon_n^2 \sigma_n^2 \tag{6-53}$$

By substituting (37) into (53), the distortion becomes an explicit function of the $\alpha_i^{(n-1)}$. For

the average rate B the distortion has two parts

$$D^{(n-1)} = \left(\sum_{i=1}^{n-1} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i^{(n-1)}}\right) 2^{-2\left(\frac{nB}{n-1}\right)} + \epsilon_n^2 \sigma_n^2 \tag{6-54}$$

$$D^{(n-1)} = \Theta^{(n-1)} 2^{-2\left(\frac{nB}{n-1}\right)} + \epsilon_n^2 \sigma_n^2 \tag{6-55}$$

The first part is an exponential decay whose magnitude is

$$\Theta^{(n-1)} = \sum_{i=1}^{n-1} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i^{(n-1)}} \tag{6-56}$$

The second part is a distortion "offset" based upon the unquantized n-th coefficient.

To get the distortion function for the j-th rate interval (44), the n−1 of (55) and (56) is replaced by n−j, and the summations are regrouped,

$$D^{(n-j)} = \left(\sum_{i=1}^{n-j} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i^{(n-j)}}\right) 2^{-2\left(\frac{nB}{n-j}\right)} + \sum_{i=n-j+1}^{n} \epsilon_i^2 \sigma_i^2 \tag{6-57}$$

$$D^{(n-j)} = \Theta^{(n-j)} 2^{-2\left(\frac{nB}{n-j}\right)} + \Phi^{(n-j)} \tag{6-58}$$

where the constants $\Theta^{(n-j)}$ and $\Phi^{(n-j)}$ are

$$\Theta^{(n-j)} = \sum_{i=1}^{n-j} \epsilon_i^2 \sigma_i^2 2^{-2\alpha_i^{(n-j)}} \tag{6-59}$$

$$\Phi^{(n-j)} = \sum_{i=n-j+1}^{n} \epsilon_i^2 \sigma_i^2 \tag{6-60}$$

Equation (59) is assumed zero for j=n, and (60) is assumed zero for j=0. Within each of the corresponding rate intervals of (44), these curves decay exponentially and are biased by $\Phi^{(n-j)}$, the distortion value of the coefficients coded at zero rate.

Figure 6.4 shows the distortion function of (58) for the four coefficient example of this chapter. (Compare this with Figure 6.5 where the Huang and Schlutheiss distortion functions are depicted.) Notice that the semilog summand functions are piecewise continuous straight lines over each of the intervals of constant activity. The distortion of (58) consists of n

summands

$$D^{(n-j)} = \sum_{i=1}^{n-j}\left(\epsilon_i^2\sigma_i^2 2^{-2\alpha_i^{(n-j)}}\right)2^{-2\left(\frac{nB}{n-j}\right)} + \sum_{i=n-j+1}^{n}\epsilon_i^2\sigma_i^2 \tag{6-61}$$

$$D^{(n-j)} = \sum_{i=1}^{n}D_i^{(n-j)} = \sum_{i=1}^{n-j}\Theta_i^{(n-j)}2^{-2\left(\frac{nB}{n-j}\right)} + \sum_{i=n-j+1}^{n}\Theta_i^{(n-j)}\epsilon_i^2\sigma_i^2 \tag{6-62}$$

where each summand is either a flat line

$$D_i^{(n-j)} = \Theta_i^{(n-j)}\epsilon_i^2\sigma_i^2 \quad \forall i \le n-j \tag{6-63}$$

or a semilog straight line

$$\log_2 D_i^{(n-j)} = \log_2\left(\Theta_i^{(n-j)}\right) - \left(\frac{2n}{n-j}\right)B \quad \forall i > n-j \tag{6-64}$$

whose slope is $-2n/(n-j)$. The only rates for which the total distortion is a semilog straight line are those for which all of the non-negativity constraints are inactive ($B \ge \alpha_n^{(n)}$). For any other rate at least one of the non-negativity constraints is active and $\Phi^{(n-j)}$ of (58) is not zero. Therefore, the summands of (58) cannot be grouped to form a straight semilog function

$$D^{(n-j)} = K_1 2^{K_2 B} \quad (K_1, K_2 \text{ are constants}) \tag{6-65}$$

In reality, equation (58) represents an upper bound for the distortion-rate function. This can be shown by considering its value at B=0. Here, all of the constraints are active, and (58) becomes

$$D = D^{(0)} = \Phi^{(0)} = \sum_{i=1}^{n}\epsilon_i^2\sigma_i^2 \tag{6-66}$$

but, the zero rate distortion is equal to the sum of the coefficient variances

$$D = \sum_{i=1}^{n}\sigma_i^2 \tag{6-67}$$

The distortion of (58) represents the zero-rate coding condition (that is, if the rate is zero the source signal is assumed to be zero) compatible with the work done within this dissertation. For $D^{(0)}$ to satisfy (67), the ϵ_i^2 must be 1. This condition is only satisfied for gaussian statistics. When using gaussian statistics, the rate-distortion bound shows that $\epsilon^2=1$ for all rates and (67) represents the true distortion function for all possible source coding rates. Since the pdfs for the

non-dc coefficients of natural images are more likely to be laplacian [77], where $\epsilon_{lap}^2 > 1$ [73], this condition is never met. Therefore, at low rates (67) can only be considered an upper bound for the actual distortion performance. The rate-constant ϵ_i^2 predicts the actual quantizer performance for large rates, this bound does becomes tight for large rates. In the next section, solutions are formulated for the case where the performance factors are allowed to vary with rate.

6.6 Distortion function using variable performance factors

In this section the distortion-rate problem of (16) and (17) are considered for the case where the performance factors are allowed to vary with rate. For this case, the derivatives of the performance factors of (21) are not zero

$$\sigma_i^2 2^{-2B_i}\left\{(-2\ln 2)\epsilon_i^2 + \frac{\partial \epsilon_i^2}{\partial B_i}\right\} + \lambda = 0 \quad \forall i \tag{6-21}$$

The Lagrange multiplier, which is the same for all n equations,

$$\lambda = \sigma_i^2 2^{-2B_i}\left\{(2\ln 2)\epsilon_i^2 - \frac{\partial \epsilon_i^2}{\partial B_i}\right\} = g_i'(B_i) \quad \forall i \tag{6-68}$$

must solve n first order differential equations for a given B. As was shown above, when the derivatives of (21) are zero, λ can be found by using the rate constraint (17a). If the $g_i'(B_i)$ are strictly monotonic, Segall [59] uses the dual optimization method (Luenberger [78]) to solve for λ, and by using the invertibility of $g_i'(B_i)$ found solutions for the coefficient rates, B_i. Segall only studied the case for gaussian sources for which the $g_i'(B_i)$ are strictly monotonic and the same for all i. When solving (68) for nongaussian sources, $g_i'(B_i)$ does not necessarily retain the monotonic property. In the general case, not all of the $g_i'(B_i)$ are the same. Both of these facts limit the applicability of his method.

Figure 6.6 shows λ as a function of coefficient rate for a laplacian pdf. Here each $g_i'(B_i)$ is invertible over only a limited range. Outside of these ranges the same value of λ is mapped into at least two different coefficient rates voiding the uniqueness of the inverse functions of (68).

Contrast this with the λ curve of Figure 6.6 where rate-constant performance factor is used, (in this figure the asymptotic laplacian performance, ϵ^2=4.7, is used). Here the Lagrange multiplier

$$\lambda = g_i'(B_i) = -2\ln 2\epsilon^2\sigma_i^2 2^{-2B_i} \quad \forall i \tag{6-22a}$$

is strictly decreasing. In fact, as is shown in the figure is a straight semilog function

$$\log_2\lambda = \log_2 g_i'(B_i) = \log_2(-2\ln 2\epsilon^2\sigma_i^2) - 2B_i \quad \forall i \tag{6-22b}$$

Now the coefficient rates can be found by inverting the $g_i'(B_i)$ functions. This gives the same solutions as demonstrated by the rate equations of (43).

The noninvertibility of (68) can prevent one from finding a closed solution for the system of (16) and (17). Computer minimization has been utilized to find numerical solutions for the coefficient rates and distortion of this system using the laplacian pdf four coefficient example. The distortion curves for this system are shown in Figures 6.7 and 6.8. Notice that as the rate increases, the solution using rate-variable performance factors approach those using the asymptotic high-rate performance factors. Recall that the solutions using the high-rate asymptotic performance factors are invertible. At high rates the rate-variable solutions are also invertible. This is true since the derivatives of the scalar laplacian pdf performance factors go to zero at high rates,

$$\frac{\partial\epsilon_i^2}{\partial B_i} \to 0 \text{ as } B_i \to \infty \quad \forall i \tag{6-69}$$

so the rate decay, 2^{-2B_i}, will dominate $g_i'(B_i)$. In fact, at high rates $g_i'(B_i)$ of (68) has the same form as (22) and becomes invertible. This is true within a portion of the rate interval, $B \geq -\alpha_n^{(n)}$, where all of the inequality constraints are inactive. Therefore, the coefficient rates for the system of (68) are of the form

$$B_i \to B + \alpha_i^{(n)} \text{ as } B \to \infty \quad \forall i \tag{6-70}$$

where the $\alpha_i^{(n)}$ are those defined in (33).

Another approach to find the coefficient rates of this section is discussed in Appendix 2.

In the appendix, the solution to the differential equations of (19) are approached using the method of exact differential equations. The solution found does not agree with those shown above. This is believed to be the results of solving a set of equations that are not guaranteed to have a unique solution. This problem is left as an open subject for future consideration.

In conclusion, consider four distortion-rate functions for the four coefficient laplacian pdf example system of this chapter. In Figures 6.7 and 6.8, these functions indicate the distortion-rate functions for these system:

1) Huang and Schultheiss upper bound using scalar quantized rates with rate-constant laplacian performances factors, ϵ^2, (Figures 6.2 and 6.5),

2) distortion upper bound using non-negatively constrained rates coded with scalar quantization and rate-constant laplacian performance factors (Figures 6.3 and 6.4),

3) distortion performance using non-negatively constrained rates coded with α-mixture scalar quantization assuming rate-variable laplacian performance factors, and

4) distortion lower bound using non-negatively constrained rates assuming the Shannon lower bound laplacian performance factors (Figure 6.9).

The first upper bound uses the coefficient rate solutions resulting from the method of Huang and Schultheiss (section 6.3) where the coefficient rates are allowed to take negative values and the quantizer performance factors use the high-rate asymptotic values. The second upper bound curve is similar to the first curve except the coefficient rates are constrained to be non-negative. The Huang and Schultheiss upper bound is less than the non-negative upper bound at lower rates because it reflects the unrealistic use of coefficient rate allocations that are negative.

The third curve uses the non-negativity constraints and the rate-variable performance factor solution of section 6.4. This distortion-rate curve uses the realizable mix of laplacian scalar quantizers indicated by (29). Since performance factors indicated by (29) are obtainable, this distortion-rate curve is also obtainable in practice. The first two upper bound curves are

asymptotic to this curve, but are not tight upper bounds. This third curve substantially lowers the upper bound estimate that is obtained by the other two bounds at low rates.

To demonstrate that the third function a performance curve was obtained using computer simulation. The performance factors of (29) were obtained using a random number generator to select which quantizer is used to code a given source symbol for nonintegral coding rates. Let x_i be the value of the i-th element of a uniformly distributed random sequence that takes values from 0 to 1. Then for rates $R\varepsilon(B,B+1)$, if $B+x_i<R$ the i-th source point is coded with an quantizer of rate B+1. If $B+x_i \geq R$ the source point is quantized with rate B. This method will code a source with a α-mixture of the end point quantizer rates. The final coding rate is $R = B+\alpha = (1-\alpha)B+\alpha(B+1)$.

For each rate tested, 100,000 independently distributed laplacian source vectors were quantized using the example system used throughout this chapter. The quantizers were optimal laplacian quantizers. Source coding rates at 0.25 bits/sample steps where tested. The results of this simulation are show in Figure 6.10. The theoretical curve does predicts the simulation results accurately.

The lower bound distortion-rate curve uses the Shannon lower bound for the laplacian pdf performance factor (lower bound $\epsilon_*^2=0.865$ [81]). This curve is generated using the non-negatively constrained rate method of this chapter. Using these constraints, the associated rate-distortion function is a lower bound for the other three mentioned above. No other laplacian-sourced quantizer can out perform this bound.

In the next chapter the ideas of this chapter are extended to include vector quantization and the emphasis is placed upon building a distortion-rate function that can be used to model the performance of MBC and MBC/PT.

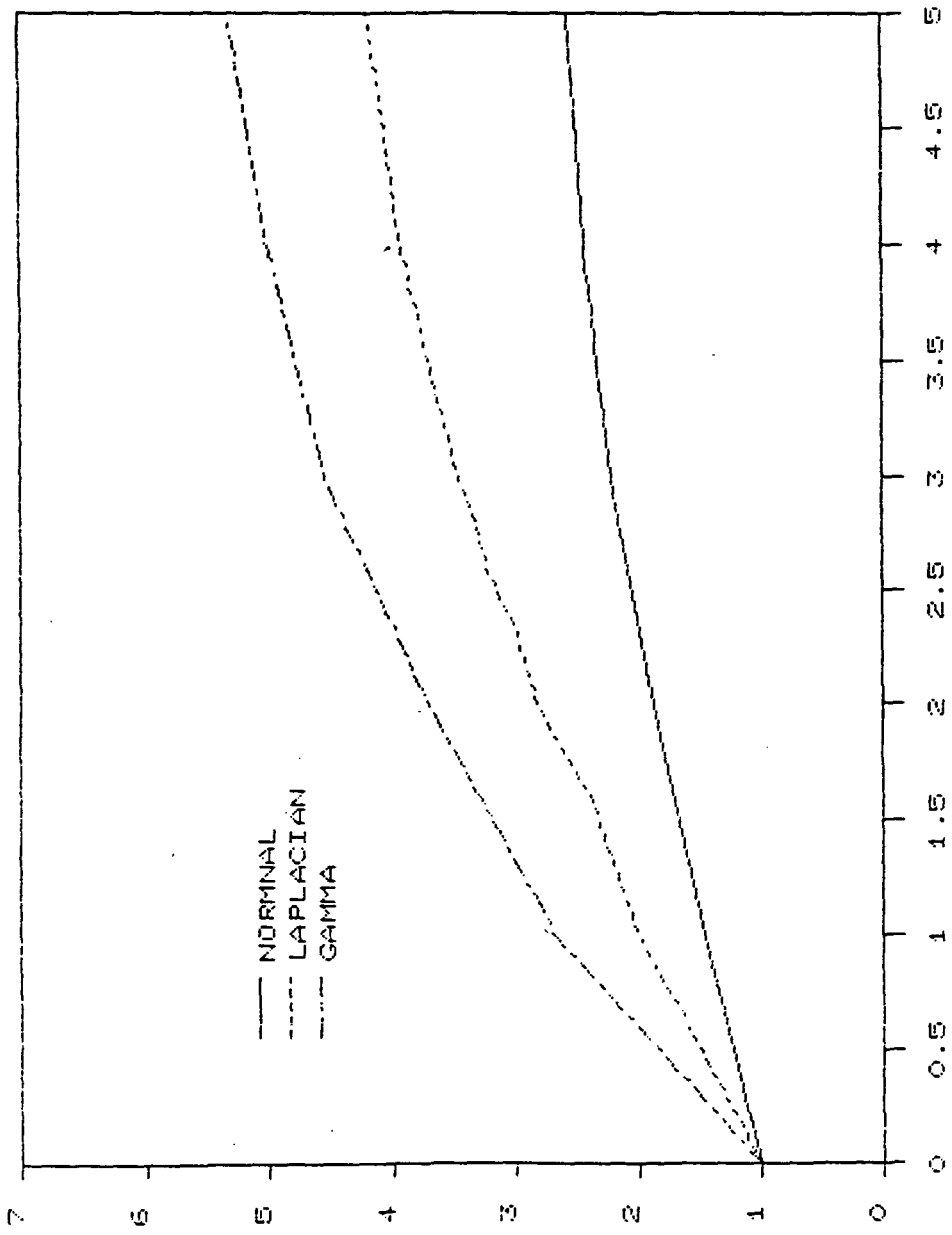


Figure 6.1 Scalar quantizer performance factors.

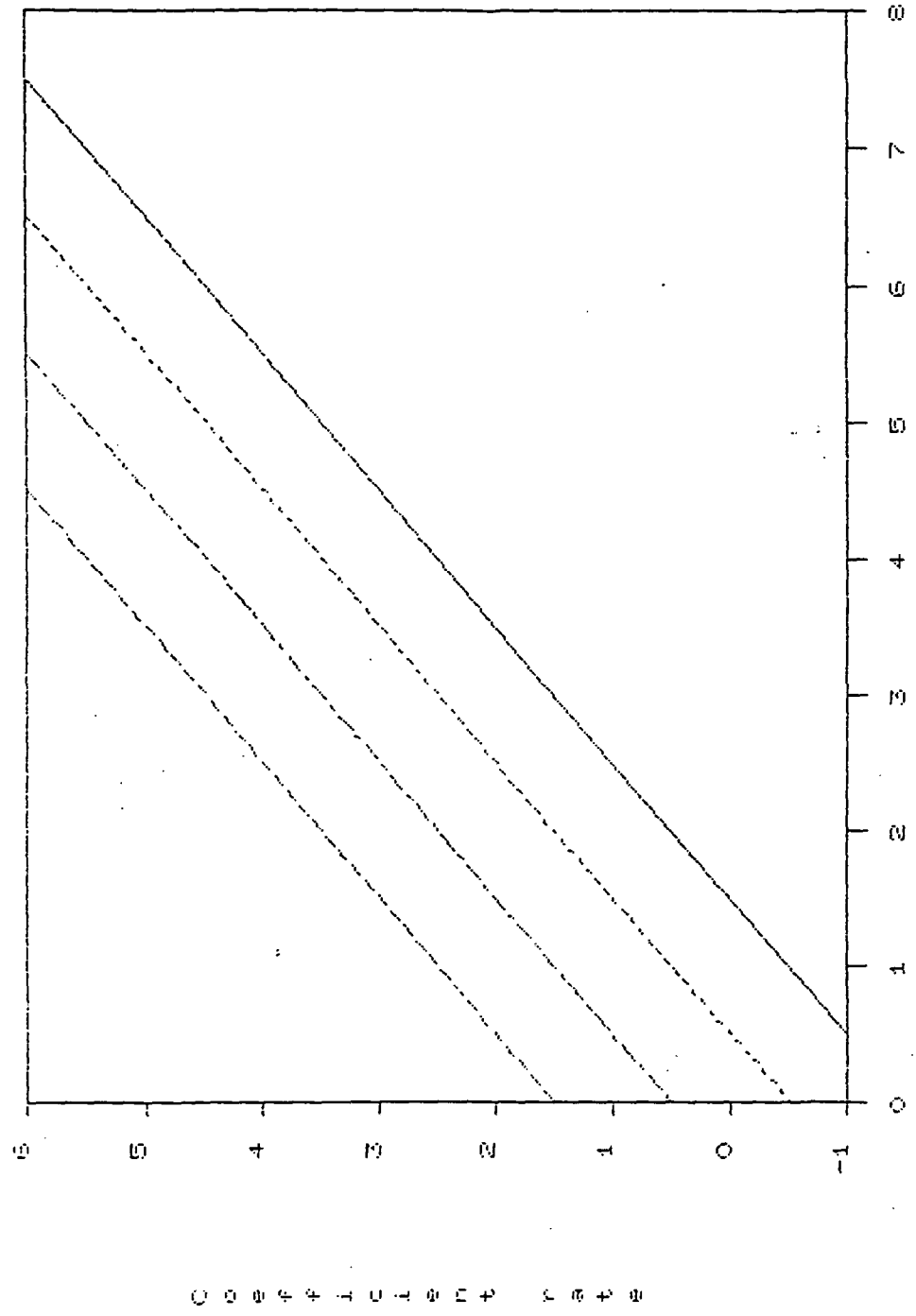


Figure 6.2 Coefficient rates using the Huang and Schultheiss solution.

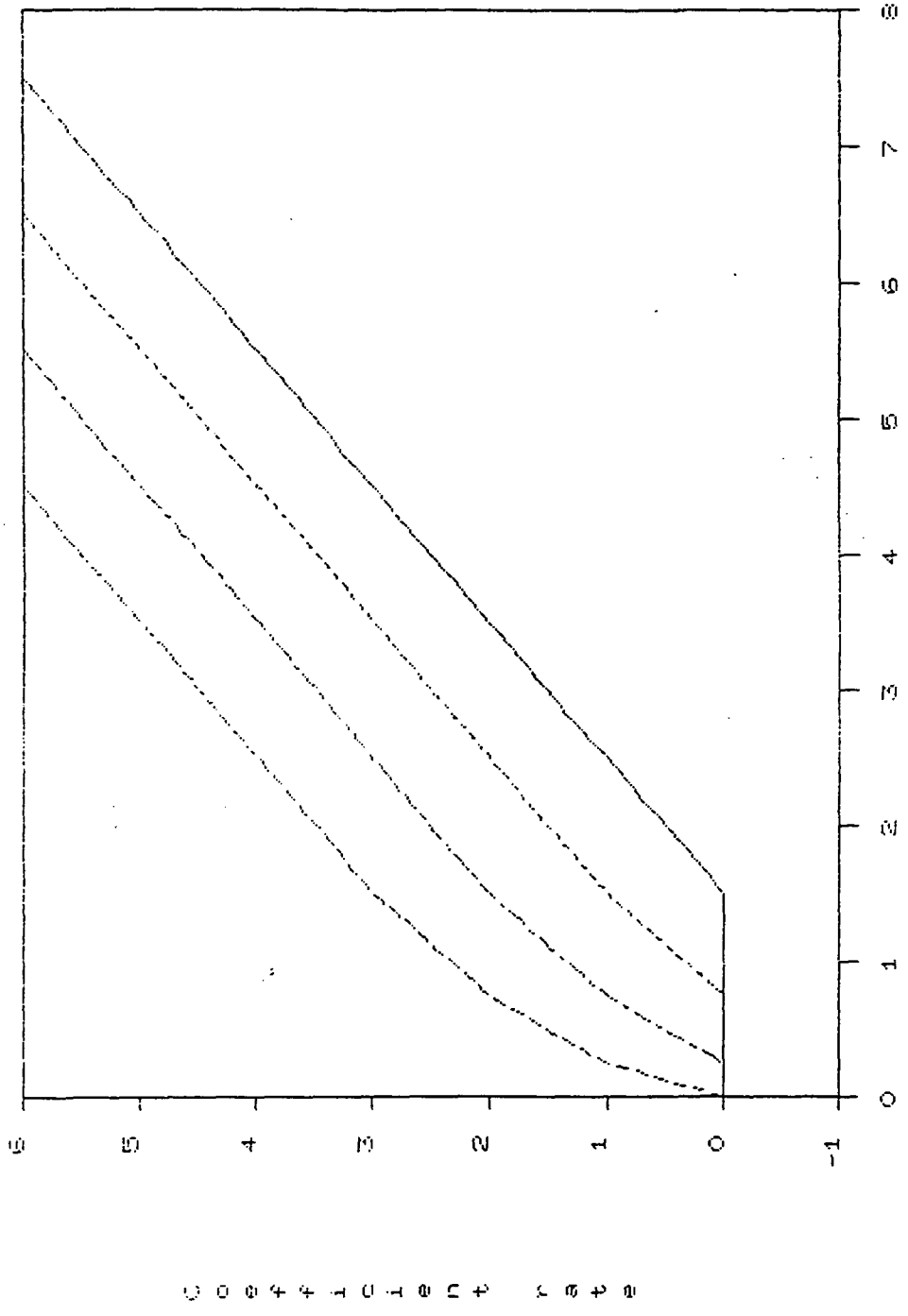


Figure 6.3 Non-negatively constrained coefficient rates with constant performance factors.

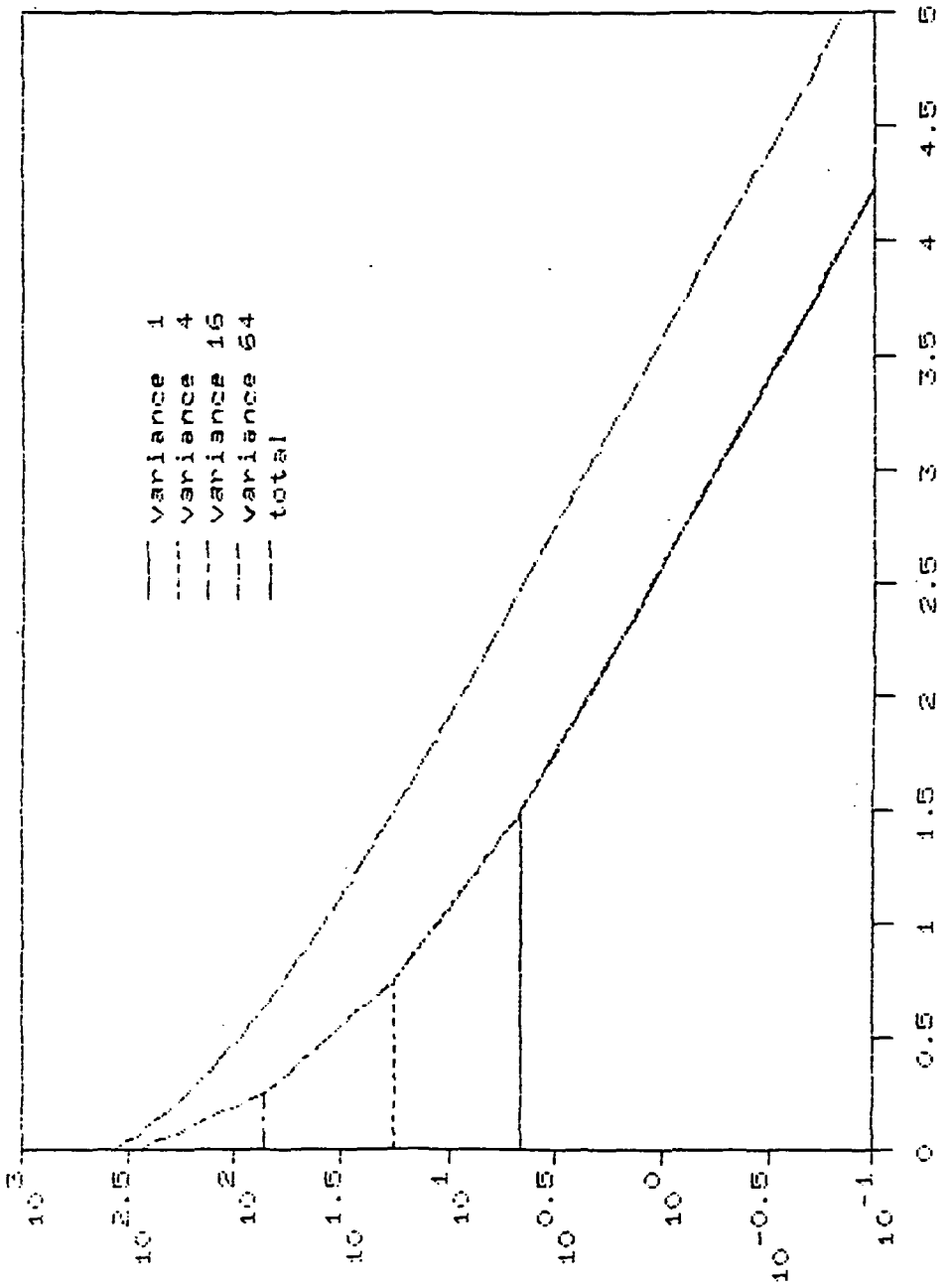


Figure 6.4 Distortion for the rates of Figure 6.3.

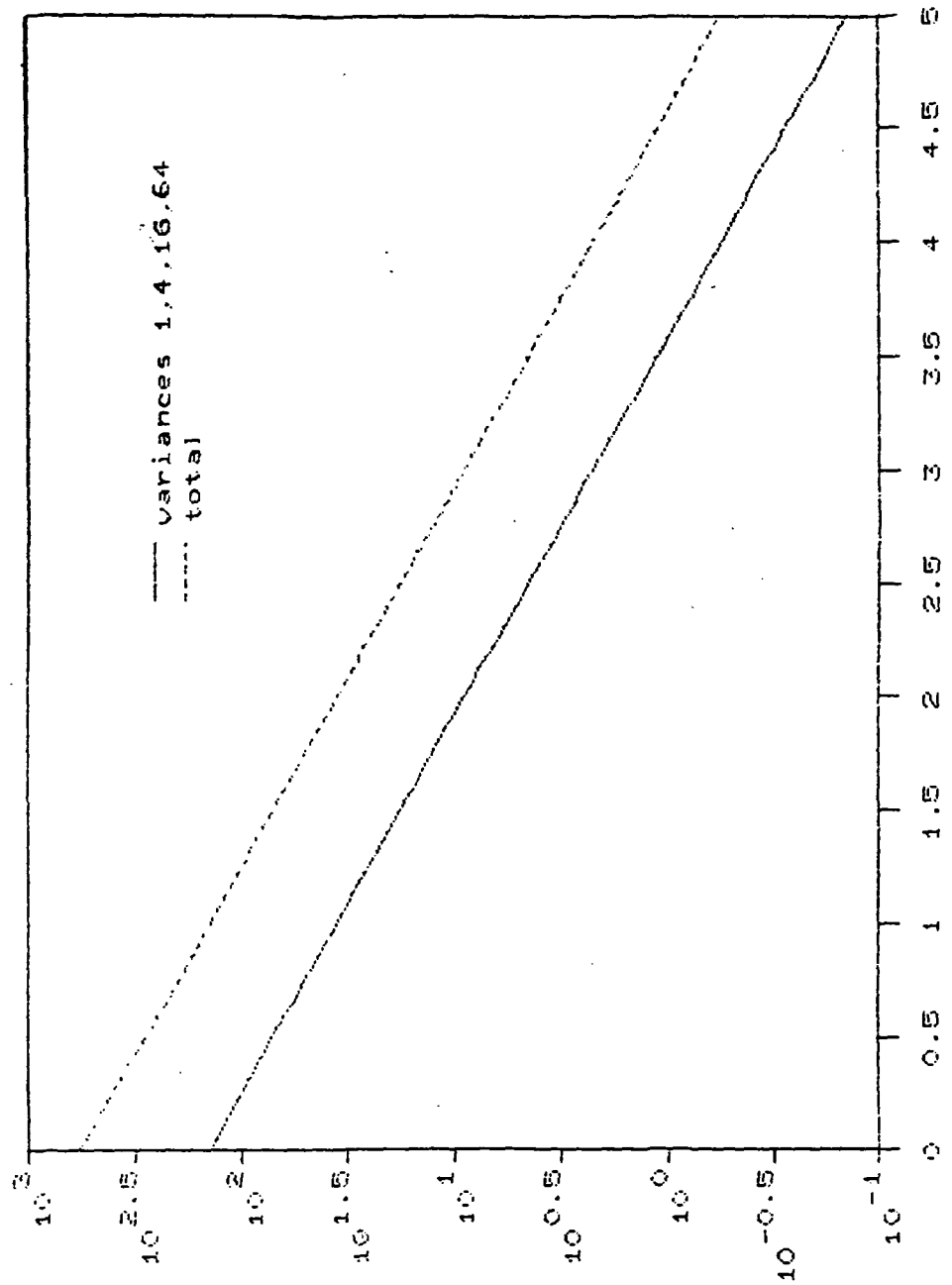


Figure 6.5 Huang and Schulthiess distortion for the rates of Figure 6.2.

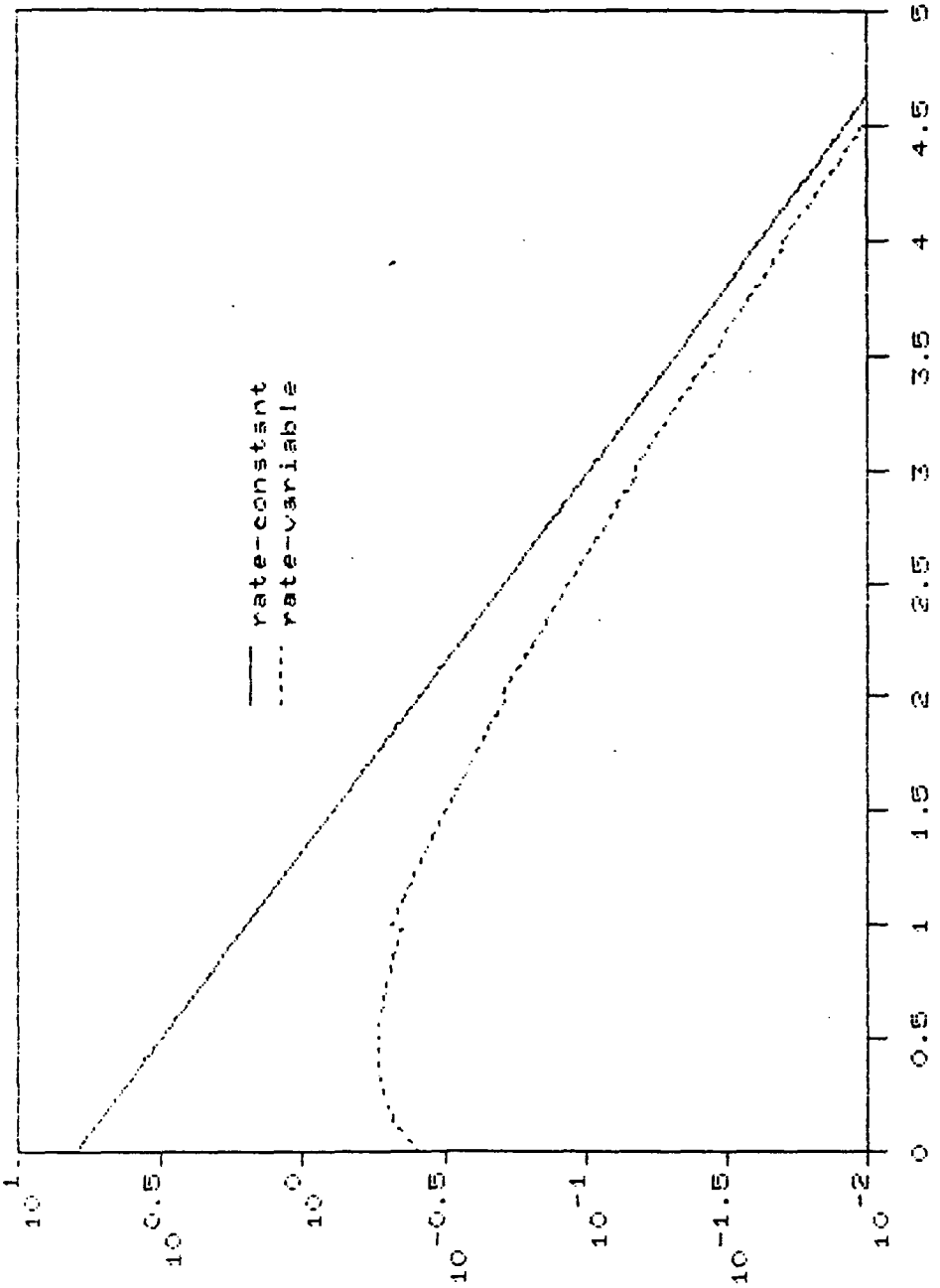


Figure 6.6 The $\log_2\lambda$ functions for the optimal laplacian scalar quantizer.

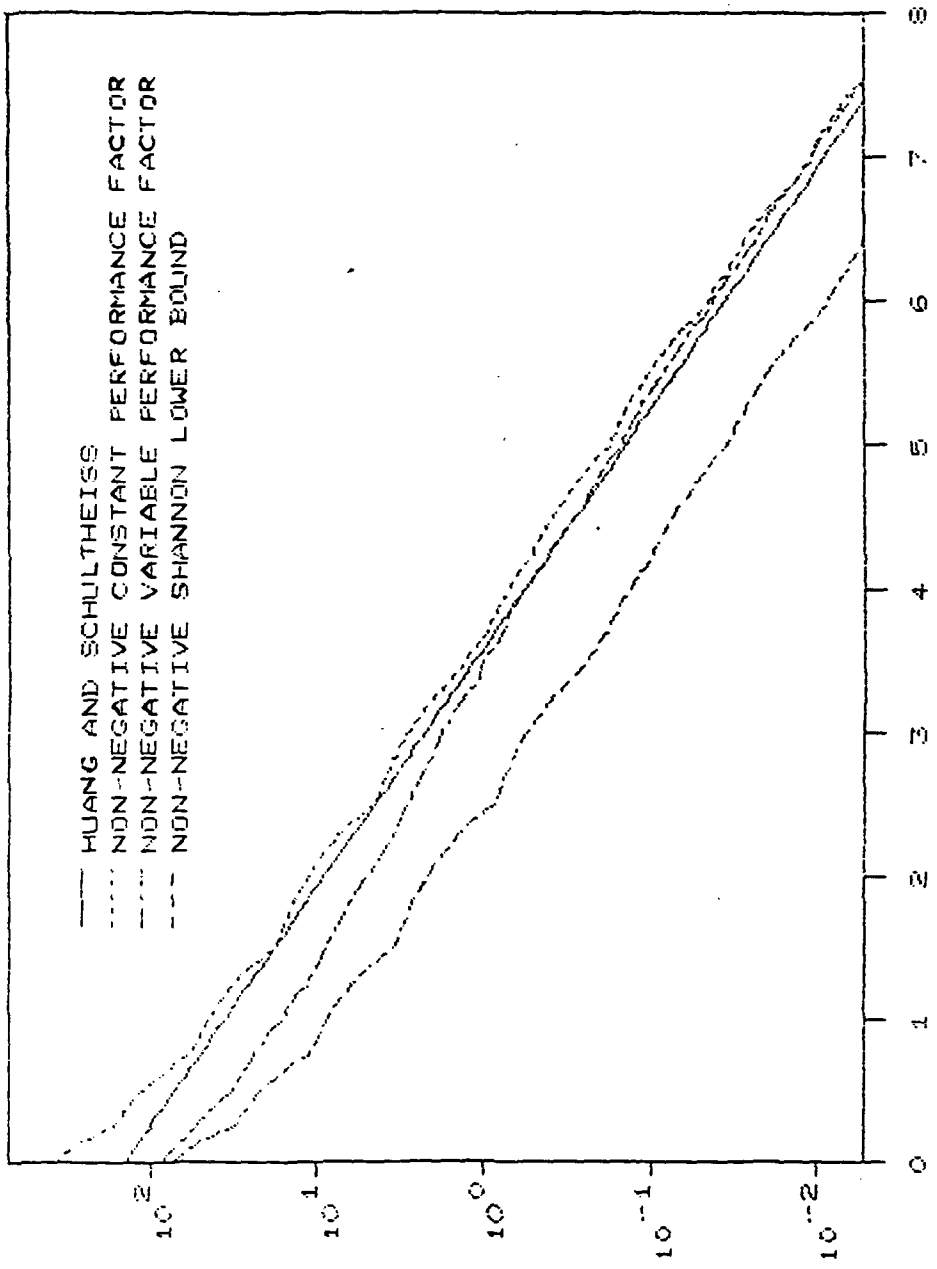


Figure 6.7 Upper and lower bound distortions for the rate solutions of Chapter 6.

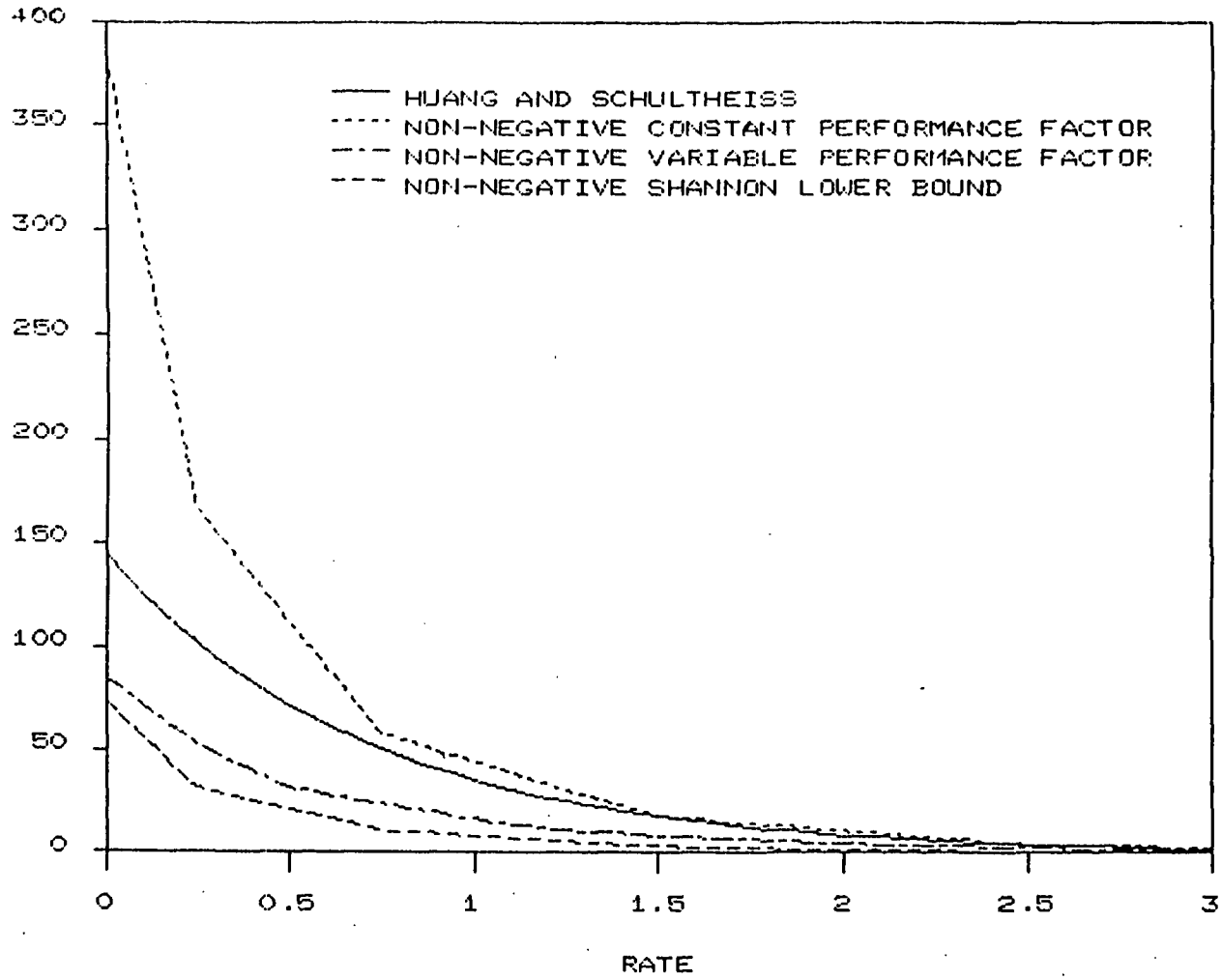


Figure 6.8 Upper and lower bound distortions for the rate solutions of Chapter 6.

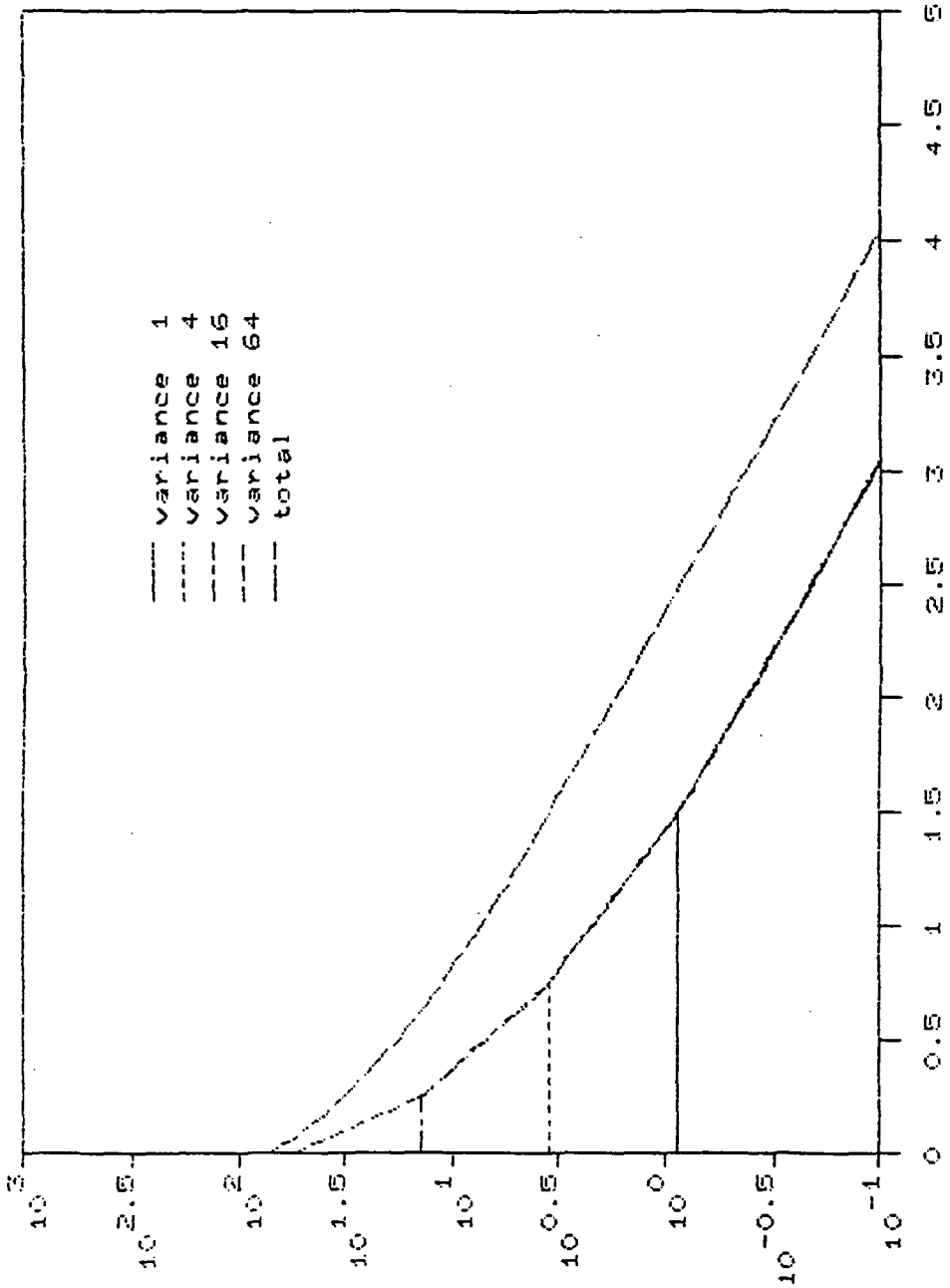


Figure 6.9 Distortion for the non-negatively constrained Shannon lower bound.

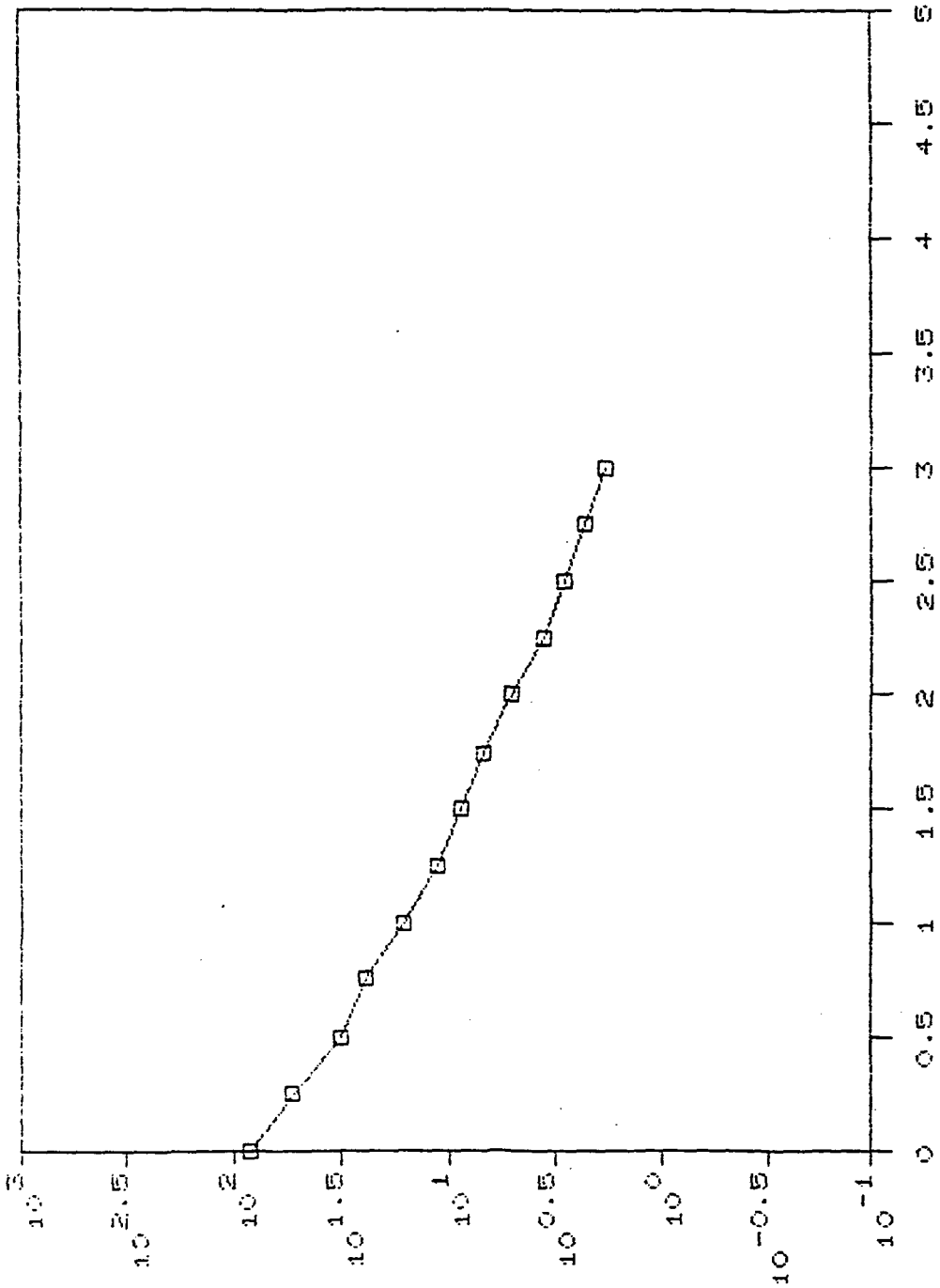


Figure 6.10 Computer simulations result using linearly weighted scalar quantizers.

Chapter 7.

A Vector Quantization Distortion-rate Function for MBC and MBC/PT

The transform source coder distortion-rate function, including the special needs of vector quantization and MBC, is the subject of this chapter. Particularly, the problem of subdividing a block of transform coefficients into sub-blocks of more manageable size is considered. A distortion-rate formulation is developed for the MBC and MBC/PT systems of Chapter 5. Some comments are also made concerning their use as a tool to explore the obtainable lower bounds for distortion of such coding schemes. Several parameter alternatives (those other than rate) for the distortion function are also developed. These distortion-rate alternates include distortion-*threshold*, distortion-*mixture-fraction* parameterizations.

7.1 Block partitioning for vector quantization

When using vector quantization techniques to code a block of transform coefficients not much information can be found in the literature to help one partition the transform block into manageable sub-blocks. In general, the entire transform blocksize is larger than can be adequately handled with a single LBG vector quantizer so the block is usually partitioned into a set of smaller sub-blocks for actual quantization. Vaisey and Gersho [66] use a system for assigning a distortion dependent sub-block partition for a variable blocksize DCT coder. But, the partitioning and the rates allocated to the vector quantizer of each block were assigned ad hoc. The goal of this chapter is to explore some of the theoretical aspects to be taken into consideration when subdividing a transform block for use with vector quantization. A distortion-rate function is developed for use with this problem.

Block distortion-rate theory shows the average distortion-rate bound for an independently distributed gaussian source of blocksize n, coded with an average rate of R, is [74]

$$D_G^{(ave)} = \left(\prod_{i=1}^{n} \sigma_i^2\right)^{1/n} 2^{-2R} = \langle\sigma^2\rangle 2^{-2R} \tag{7-1}$$

where $\langle\sigma^2\rangle$ is the geometric mean of the coefficient variances of the block. The (ave) symbol is used to indicate the distortion measured is a per coefficient average. This shows that the distortion of a coded block is proportional to the number of quantizer code points and the geometric mean of the block variances. For a non-Gaussian source this can be generalized using the Shannon lower bound [74], which incorporates a performance factor ϵ_*^2:

$$D^{(ave)} = \epsilon_*^2 \langle\sigma^2\rangle 2^{-2R} \tag{7-2}$$

This is a direct analog of the Huang and Schultheiss distortion formulation of the last chapter. If the performance factor is equal to one, $D^{(ave)}=D_G^{(ave)}$, the known solution for gaussian statistics. The total distortion of the whole block is

$$D = \sum_{j=1}^{n} \epsilon^2 \langle\sigma^2\rangle 2^{-2R} = n\epsilon^2 \langle\sigma^2\rangle 2^{-2R} = nD^{(ave)} \tag{7-3}$$

If the source is to be partitioned and coded with N disjoint sub-blocks, the total distortion of the source coder is

$$D_T = \sum_{k=1}^{N} D_k \tag{7-4}$$

If the number of elements in the k-th block is n_k, then the average distortion obtained by this block partition is

$$D_T^{(ave)} = \frac{\sum_k D_k}{\sum_k n_k} \tag{7-5}$$

or in terms of the sub-block average distortions

$$D_T^{(ave)} = \frac{\sum_k n_k D_k^{(ave)}}{\sum_k n_k} \tag{7-6}$$

So far, this discussion has been based upon the partitioning of a single block in a known fashion. But, when performing MBC, different sized blocks are mixed together. Each of which many be partitioned differently. How to formulate a distortion function to include multiple transform blocksizes is considered next.

7.2 Rate as function of the MBC mixture and thresholds

Consider the relationship existing between the coding rate and mixture fractions when using MBC. If the average coding rate of the different block sizes is known, say R_k for the k-th blocksize, then the average rate at which an image is coded is determined by the mixture fractions, p_k,

$$R = R(p_1,p_2, \ldots ,p_N) = \sum_{k=1}^{N} p_k R_k \tag{7-7}$$

The average block rate is definable as

$$R_k = \sum_{k=1}^{N} p_k \frac{B_k}{S_k} \tag{7-8}$$

where B_k is the number of bits used to code the k-th block whose size is S_k. Since the entire image is to be coded, consistency demands

$$\sum_{k=1}^{N} p_k = 1 \tag{7-8}$$

Since the sub-block mixture fractions and the total coding rate are related through (7), it is sometimes easier to compute the distortion in terms of the fractions instead of the individual block rates.

When using the MBC method, the mixture fractions are determined by the block thresholds. As was indicated in Chapter 5, the mixture fractions change with threshold (see Figure 5.5). Notice that the mixture versus threshold profile is different for different images (e.g., Figure 5.12). This indicates the coding rate will be different from image to image.

To formalize this last paragraph, let a set of n scalar mappings $\{p_k\}_{k=1}^{n}$ be defined from the scalar threshold space, $T \doteq R_+$, into the n-dimensional mixture-fraction space, $F = R_+^n$. To start with, assume that the same threshold is used for each blocksize. Then

$$p_k = p_k(t) \text{ for } t \varepsilon T, \text{ so that } (p_1(t), p_2(t), \ldots ,p_n(t)) \varepsilon F \tag{7-9}$$

Once these mappings are known, the coding rate is a function of the MBC thresholds

$$R = R(p_1(t), p_2(t), \ldots ,p_n(t)) = R(F) \tag{7-10}$$

In general, the p_k maps will vary with the image, say p_k^i for the image $X_i \varepsilon X$. Now the rate is a

function of the image, as well as the threshold,

$$R = R(p_1^i(t), p_2^i(t), \ldots, p_n^i(t)) = R(F; i) \tag{7-10a}$$

With this formulation, all images that have similar mixture maps can be classified into a single subset of the set of all images, **X**. All of these images will code at the same rate. This point is useful when studying the different classes of images, but for the remainder of this chapter it is assumed that only a single image (or image class) is to be studied. This allows us to use the simpler rate models of (10).

The distortion-rate function of mixture block coding is a mapping "from a multi-dimensional space" and it would be desirable to reduce the problem to one of a mapping "from a one-dimensional space." In reality, the function of (10)

$$R = R(p_1(t), p_2(t), \ldots, p_n(t)) = R(F) = R(t) \tag{7-10b}$$

is not multi-dimensional since the mixture maps are all a function of a single variable, the threshold. But, the formulation of (10) is an oversimplification of the problem. The reason for having a mixture coding system in the first place is to allow one to code regions of different difficulty with different block types. This must be done is such a way that allows one to partition the blocks of one blocksize independently of the way one partitions blocks of another blocksize. This can be done by allowing the thresholds of each blocksize to vary independently.

If the threshold is allowed to vary independently for each blocksize, the rate becomes a true multi-dimensional mapping from the threshold space and the mixture fraction space. This is shown by reformulating (9) and (10),

$$p_k = p_k(t_k) \text{ for } t_k \varepsilon T, \text{ so } (p_1(t_1), p_2(t_2), \ldots, p_n(t_n)) \varepsilon F \tag{7-11}$$

$$R = R(p_1(t_1), p_2(t_2), \ldots, p_n(t_n)) = R(t_1, t_2, \ldots, t_n)\Big|_{\{p_k\} \varepsilon F} \tag{7-12}$$

The scalar mapping property is lost completely. Since this definition is more general (less constrained) it will give a lower distortion-rate bound than will (10).

The mixture and rate maps of (10) mappings are not continuous functions of the threshold, since the mixture fractions must be taken from a discrete set. For example, a image that contains 256×256 pels can be partially coded with 16×16 block for only the fractions 1/256, 2/256, 3/256, . . . , 256/256. For the mixture space to be continuous it would be necessary to use fractional blocks. It would be necessary to add rigor to the meanings of these fractional mapping definitions, if a more complete analysis of MBC using distortion-rate theory were the goal, but this problem is not critical to the material discussed here and is left for future study.

Now that the rate is known as a function of the thresholds and mixture fractions, it is possible to use these formulations to derive a distortion-rate function that is not only a function of rate but also the thresholds and mixture fractions.

7.3 Threshold driven distortion-rate functions

In section 7.1 a method for coding a block using a known mixture of vector quantizers was discussed. This material is directly applicable to the material presented here. When taken with the threshold-rate determinations of the last section, a distortion-rate function, or more conveniently a threshold-distortion function, can be obtained for a given threshold value.

The average distortion of an MBC coder is defined by

$$D_T^{(ave)} = \frac{\sum_k p_k D_{T_k}}{\sum_k p_k n_k} \tag{7-13}$$

where D_{T_k} and n_k are the distortion and block size for the k-th MBC block type. The summation is taken over the number of blocksizes used by the MBC coder. The D_{T_k} distortion values are assumed to be vector-quantized transform-coded blocks as discussed above. They take the form indicated by (4)

$$D_{T_k} = \sum_{i=1}^{N_k} D_{k,i} \tag{7-14}$$

where N_k is the number of sub-block partitions of the k-th block type, and $D_{k,i}$ are the different distortion totals for each of the vector-quantized sub-blocks of the k-th MBC block type.

Notice the denominator of (13) is equal to the number of pels in the image. If the number is equal to N×N, (13) becomes

$$D_T^{(ave)} = \frac{\sum_k p_k D_{T_k}}{N \times N} \tag{7-15}$$

Since N×N is a constant, the distortion function can be redefined, without loss of generality, as

$$D_T = \sum_k p_k D_{T_k} \tag{7-16}$$

The D_{T_k} distortions can be found using any source coding method. (This form of the equation assumes that the total distortion is the sum "expected" distortion terms. This feature is not necessary true, and will be addressed at the end of the chapter.) They do not have to be computed using the specialized vector-quantized transform-coded MBC methods emphasized in this dissertation. The final coding rate obtained is a function of these particular block coders and it may be desirable to use a different coding method for each blocksize.

The MBC distortion is to be minimized given the rate or, equivalently, the mixture fractions. Also, the distortion can be minimized using scalar or vector threshold parameterization. So the performance of an MBC coder can be parameterized by one of four methods:

$$\min_R D_T, \quad \min_{\{p_k\}} D_T, \quad \min_t D_T, \quad \min_{\{t_k\}} D_T \tag{7-17a-d}$$

Not all of these functions give the same distortion function. The minimization (17a) represents the general formulation. Equations (17b) and (17d) are just reformulations identical to (17a). These formulations reside in the same multi-dimensional solution space no matter which parameterization is used. Only (17c) resides in a singly dimensioned space. Since (17b) is more severely constrained than the other formulations it is an upper bound for them.

Once the form of the D_{T_k} are known the minimizations of (14) can be done. This is dependent upon the type of block coders used and is the subject of the next section.

7.4 Distortion-rate function for MBC

The distortion function for the vector-quantized MBC systems described in Chapter 5 is described here. The different blocksizes[1] for this method are all coded similarly using a DCT. Only four coefficients are coded per block and these coefficients are divided into two sub-blocks for quantization. This accounts for a three-way block partition. The first partition sub-block contains only the dc DCT coefficient. The second contains the next three lowest frequency DCT coefficients. The last contains the remaining DCT coefficients, all of which are not coded. The block distortion function (14) is the same for all blocks and is of the form

$$D_{T_k} = \sum_{i=1}^{3} D_{k,i} \tag{7-18}$$

The dc coefficient codes with distortion that must take into consideration the possibility of a non-zero mean

$$D_{k,1} = \epsilon_{k,dc}^2 \mu_{k,dc}^2 2^{-2B_{k,1}} \tag{7-19}$$

where the $\mu_{k,dc}^2$ are not necessarily the coefficient variances. If the dc coefficients are non-zero and this must be included in the model used. The second sub-block distortions are

$$D_{k,2} = \epsilon_*^2 (\sigma_{k,2}^2) 2^{-2B_{k,2}} \tag{7-20}$$

and the remaining uncoded coefficient distortions are

$$D_{k,3} = \sum_{l \in \mathcal{B}_k} \sigma_l^2 \tag{7-21}$$

---

[1]Up to this point in the text of this chapter, the coefficients of a block have been assumed to be scalar valued. In general, the coefficients can be vectors, as is the case when coding a color image. When this is the case, the block has three times as many coefficients and problems can arise when one counts the number of entities being coded within a block. In the text it is assumed that the coefficients are scalar and all totals reflect this fact. If a need to study vectored coefficients is important, the numerical totals used to count how many coefficients occur within a block need to updated accordingly.

The symbol $\langle\sigma^2_{k,2}\rangle$ represents the geometric mean of the three non-dc coefficients to be vector quantized in the second sub-block of the k-th MBC block type. The $B_{k,i}$ represent the average coding rate allocated for the i-th sub-block of the k-th MBC block type. In the last equation, $\mathfrak{B}_k$ represents the index set for the uncoded coefficients of the k-th MBC block type. For the four coefficient DCT coding $\mathfrak{B}_k$ contains n_k^2-4 coefficients. Notice if the block size is 2×2 then $\mathfrak{B}_k$ is vacuous and $D_{k,3}=0$.

All of the non-dc coefficients can be assumed to have laplacian distributions. Therefore, the vector quantizer performance factors can be modelled using the Shannon lower bound for laplacian sources. The dc coefficients are not as easily modelled. Some have suggested that various unimodal pdfs can be used to model transform-coded dc coefficients, but natural images are, at many times, multimodal and very difficult to model parametrically. This problem must be left open unless more information can be found for the particular images being coded.

The magnitudes of the uncoded variances of (21) are hard to model. Not much work has been done to study this problem, except to say these coefficients can be modelled with a laplacian pdf that is parameterized by the coefficient indexes. One method uses a sum of the coefficient indexes [14]

$$p(i_x,i_y) = \exp\{-(\alpha_x i_x+\alpha_x i_x)\} \quad i_x,\ i_y = 0,\ 1,\ \ldots,\ N\text{-}1 \tag{7-22}$$

where i_x and i_y are the 2-dimensional coefficient indexes of the N×N block, and the α_x and α_y are constants. Here the covariance model is separable. Other work has been done to study specific problems, but very little can be added to the study of the more general problem. Most models, including the one mentioned above, assume stationarity. Since the work done here is an attempt to overcome the problems associated with nonstationarity, these facts are of little use.

One way to overcome this problem is to classify the various type of images by their block percentage versus distortion threshold profiles, as was mentioned in conjunction with equation (10a). Of course this assumes that the block variances can be modelled more accu-

rately using this profile classification system. It is reasonable to assume the block variance "profile" can be modelled similarly for all images having similar block percentage profiles. This assumption has not been validated. This is partially due to the fact that the available image database is not large enough to adequately pursue the subject.

With the last consideration held aside, the total MBC distortion as represented by (17) is now known, and once the type of parameterization is selected (e.g., coding rate, block percentages, or distortion thresholds) the minimization process can be pursued.

Since the distortion function of this chapter is represented by a sum of variance-weighted exponential decays, they differ from those used in the last chapter only in one significant point that is easily handled. The uncoded coefficients "trail along" from one pass to the next acting as an offset to the overall distortion baseline which is similar to the coefficients that are coded with zero rate in Chapter 5.

In review, the same approach can be used to solve for the distortion function and the associated coefficient rates for a MBC system as were used in Chapter 5. The only important difference comes with the inclusion of the more general nature of performance factors that model vector quantized blocks. But, this problem is one of modelling performance factors not with the definition rate-distortion functions. In the next section, the MBC distortion function developed above section is modified to include the effects of progressive transmission.

7.5 Distortion-rate function for MBC/PT

The distortion function for a MBC/PT system can be modelled in nearly the way as was developed in the last section. When using MBC with progressive transmission, the image is coded in a sequence of passes. The algorithm codes each pass with smaller blocks whose coefficients are generated by using the residuals of the previous pass. Interestingly, the residual non-dc coefficients for the DCT based MBC/PT computer simulations done for this dissertation

showed that the residuals are nearly laplacian, as were those found for MBC computer simulations. After the first pass, the laplacian performance factors can be used to estimate the coding performance for the three DCT ac coefficients coded in these simulations. (If more coefficients are coded, their performance factors would need to be determined.)

Since the coding of the first-pass dc coefficients is the same as has been discussed above, only the distortion summands that correspond to the dc coefficient residuals of the subsequent passes need to be reformulated. Through simulation it was found that these subsequent-pass dc-coefficient residuals are modellable as a zero-mean laplacian process. The variance of the dc residuals were found to be somewhat less than for the ac residuals of the same blocksize. This fact was compensated for in the computer simulations of Chapter 5 by using an additional scaling factor.

To reflect these changes the MBC/PT distortion of (19) needs to be reformulated. The first pass distortion function remains the same

$$D_{1,1} = \epsilon^2_{1,dc}\mu^2_{1,dc}2^{-2B_{1,1}} \tag{7-19a}$$

but, the subsequent passes need to be modelled differently

$$D_{k,1} = \epsilon^2_{*}\sigma^2_{k,\nabla dc}2^{-2B_{k,1}} \quad \forall k>1 \tag{7-19b}$$

where $\sigma^2_{k,\nabla dc}$ is the variance of the dc coefficient residuals that are passed to subsequent passes of the coder.

Other than this one change to (19), equations (17) through (21) represent the distortion function for any image coded using an MBC/PT system. Specifically, for the three block partition scheme used above, the k-th blocksize distortion function is

$$D_{T_k} = \sum_{i=1}^{3} D_{k,i} \tag{7-23}$$

where the dc coefficients (i=1) are defined by (19a) and (19b), the coded ac coefficients (i=2) are modelled with an exponentially weighted distortion function

$$D_{k,2} = \epsilon^2_{*}(\sigma^2_{k,2})2^{-2B_{k,2}} \tag{7-24}$$

and the uncoded ac coefficients (i=3) are unchanged by the coding process

$$D_{k,3} = \sum_{l \in \mathcal{B}_k} \sigma_l^2 \tag{7-25}$$

The total distortion is found by summing (23) over the entire image where the k used for any portion of the image is based upon how many passes were used to code it. More concretely, let

$$\mathcal{P}_k = \{m | d(B_m, \overline{B}_m) < d_{min,k}\} \tag{7-25}$$

be the index set of the image blocks that pass the k-th blocksize distortion threshold test. In (25), the $B_{m,k}$ are image blocks of the k-th coding pass and the $\overline{B}_{m,k}$ are their coded representations.[2] Then the total image MBC/PT distortion is similar to (16)

$$D_T = \frac{1}{\sum_i |\mathcal{P}_i|} \sum_{k=1}^{N} \sum_{m \in \mathcal{P}_k} D_{T_k}^{(m)} \tag{7-26}$$

where the (m) is used to indicate the distortion obtained for each block as it is coded using (23). The percentages, p_k, of (16) that weight the expected block distortion terms have been replaced by the image partition index sets.

As was mentioned above, the notation used in (16) forces the D_{T_k} to be formulated using either statistical expectations or non-parametric "averages" of the block distortion functions. In general, this may not be correct. The coding distortion of any block (or subset of blocks) taken from an image may be very different from the average taken over all of the blocks of the image. This fact destroys the stationary assumption used to formulate (16). Equation (26) overcomes this problem by generalizing (16) to the distortion for each image block to be

---

[2]Equation (25) includes an abbreviated form of the distortion threshold function introduced in Chapter 5. For example, if the same maximum and minimum blocksizes used in Chapter 5 are assumed, then

$$d_{min,1} = d_{min}(16\times16),$$

$$d_{min,2} = d_{min}(8\times8), \ldots, \text{and}$$

$$d_{min,4} = d_{min}(2\times2).$$

functionally different. Since the $\mathcal{P}_k$ definition used in (26) is also valid for MBC, this generalized form of the distortion function can also be used to model MBC systems.

In conclusion, several distortion functions have been described in this chapter; one set can be used to describe the distortion obtained for MBC systems and the other for MBC/PT systems. Both of which can be described using the generalized equation (26). Each of these distortion functions can be explicitly parameterized using any of the four forms of (17). The specific form used depends upon which parameterization (rate R, mixture fractions $\{p_k\}$, block thresholds $\{t_k\}$, or equivalent image threshold t) is of the most use to the particular distortion study to be made. Any of these formulations are general enough to include the image blocks that are coded using either scalar quantization or vector quantization. With the appropriate modifications needed to include the uncoded transform coefficients of the different blocksizes these functions are direct extensions of the distortion functions developed in Chapter 5. The results of this chapter can be similarly extended to included the formulation of optimal coefficients rate assignments by using the methods demonstrated in Chapter 5.

Chapter 8.

Summary and Conclusions.

Chapters 1 through 4 are include to provide background material in fields of scalar and vector quantization, transform coding, visual and CRT effects, and progressive transmission over low-bandwidth channels. These areas of review are useful in the development of the mixture block coding techniques of Chapter 5, and the theoretical material of Chapters 6 and 7.

The major results of this dissertation are presented in Chapters 5, 6 and 7. In Chapter 5, the details of the mixture block coding algorithm and its modification for progressive transmission developed for this dissertation were developed. Examples using MBC and MBC/PT were presented for the coding of monochrome, RGB and YIQ images.

The examples of Chapter 5 used scalar quantized MBC and MBC/PT systems to code images of these three different color types. These scalar systems were used to select the training sets from which LBG vector quantizers were built. The vector quantizers were tested in MBC and MBC/PT coders using the same images as were coded using the purely scalar quantized systems. The different quantization systems that coded these examples were designed with the particular requirements and properties of each color type used: RGB and YIQ. The common factors known to be true concerning the expected distributions of the different color planes, and results taken from preliminary computer runs, were used in this design process. The preliminary computer work was helpful in selecting the quantizer scaling factors. The same scaling factors are useful over a wide range of images.

The results of this chapter include quantitative data that demonstrates the relationship between the mixture fractions, average rate and distortion as a function of the distortion thresholds and quantizer system used. The examples complemented the theoretical development of

the early part of the chapter that explored the relationship between the mixture fractions and the intermediate and final coding rates of MBC and MBC/PT.

The results of Chapter 5 demonstrate the viability of MBC and MBC/PT as low- and medium-rate image source coders. The example coders performed well and demonstrated the fact that MBC and MBC/PT can be used with a variety of block coding methods. Block coders, including a DCT method and a BTC modified DCT method, were used in the examples. The modified BTC method deserves further testing as an alternate source coding method for the basic BTC method of Delp of Mitchell.

A theory of selecting optimal rate allocations for quantized transform coefficients was developed in Chapter 6. The standard bit allocation method (Huang and Schultheiss) commonly used in the literature was introduced as a starting point and modified to incorporate the realities of actual scalar quantizers. The need for including non-negative coefficient rate constraints and quantizer performance factors that vary with rate was discussed. How these factors change the standard rate and distortion solutions was developed and examples were given to demonstrate their effects. A distortion-rate function was developed for a mixture method for scalar quantization. This distortion-rate function was shown to be obtainable through computer simulation.

The standard distortion-rate function is an upper bound for the performance of actual source coders. It was compared with an other upper bound that included the effects of the coefficient rate non-negativity constraints. When rate-dependence of the quantizer performance factors is included, an obtainable distortion-rate function is obtained. Comparisons of these various systems were made using a four coefficient system that were laplacian distributed. The laplacian pdf was chosen to match the coefficient pdfs found in the MBC and MBC/PT systems of Chapter 5. Except at high data rates where all three upper bounds showed similar distortion functions, the non-negatively constrained rate-variable performance factor system showed less overall distortion than either of the more simple distortion-rate formulations. A lower distor-

tion-rate bound, using the Shannon lower bound, was included to help bracket the expected range of possible source coder distortion-rate functions.

The distortion-rate functions developed can be used in the study of systems whose coefficients come from differing source pdfs because the formulations of Chapter 6 included generalized performance factors that could be different for each coefficient quantized. The final sections of Chapter 5 included comments about the difficulty of finding solutions whose dual (Lagrange multiplier) is not invertible. This loss of invertibility prevents the mappings from the coefficient rate space into the dual space from being one-to-one.

In Chapter 7, the distortion-rate functions introduced in Chapter 6 were expanded to include the effects of vector quantized blocks of transform coefficients. How these functions can be used to develop distortion-rate functions for MBC and MBC/PT was included. These distortion-rate functions were formulated using the MBC and MBC/PT mixture fractions and distortion thresholds as parameters. This introduced the possibilities of studying MBC and MBC/PT systems using distortion-mixture and distortion-threshold functions instead of the standard distortion-rate functions.

The mixture versus threshold characterization of images shown in Chapter 5 offers a new method for classifying different image types. Images that have similar mixture-threshold functions should offer similar distortion functions.

In conclusion, the variable blocksize methods presented in this dissertation are viable image source coders. They overcome the inherent problems associated with the coding of natural images. The non-stationary characteristics of images are automatically compensated for by adaptively selecting the correct blocksize to code the different regions. The correct blocksize is selected using a distortion threshold algorithm. Difficult to code regions are coded with small blocks and easy regions with large blocks. By using this method, no more channel capacity is used to code any region than is necessary.

Appendix 1.

A Simple Bit Allocation Example

A coding example for assigning the optimal coding rates for a set of two coefficients is presented using the method of Huang and Schultheiss [70]. It is shown that their method can produce negative rates, and when the rates are forced to be non-negative the distortion will increase. Then results are given to demonstrate the effects of forcing the coefficient rates to be greater than zero.

Let the coefficients have variances

$$\sigma_1^2 = 1/4,\ \sigma_2^2 = 4 \tag{A1-1}$$

and let their the performance factors be equal, $\epsilon_1^2 = \epsilon_2^2$. The Huang and Schultheiss method assign the optimal coefficient rates (6-12):

$$B_1 = B + \tfrac{1}{2}\log_2\frac{\sigma_1}{\sigma_2} = B - 1 \tag{A1-2}$$

$$B_2 = B + \tfrac{1}{2}\log_2\frac{\sigma_2}{\sigma_1} = B + 1 \tag{A1-3}$$

where B is the average coefficient rate.

Consider what happens to the rate for the first coefficient when the average is less than 1 bit/coefficient:

$$B_1 < 0 \text{ for } B < 1 \tag{A1-3}$$

This is undesirable because coefficients can be coded only with non-negative rates. If we ignore this fact, the distortion using the rates of (3) be found from (6-16)

$$D(B) = \sigma_1^2 2^{-2B_1} + \sigma_2^2 2^{-2B_2} \tag{A1-4}$$

The coefficient rates for an average rate of, say, $B = \frac{1}{2}$ are $B_1 = -\frac{1}{2}$ and $B_2 = \frac{3}{2}$. The distortion at this average rate is

$$D(\tfrac{1}{2}) = \sigma_1^2 2^1 + \sigma_2^2 2^{-3} = \tfrac{1}{2} + \tfrac{1}{2} \tag{A1-5}$$

As can be seen, the distortion for the σ_1^2 coefficient is twice that obtained if this coefficient were not coded at all.

If the coefficient rates are constrained to be non-negative, the optimal rates are reassigned using (6-22),

$$B_1 = 0 \tag{A1-6}$$

$$B_2 = 2B + 0 = 1 \tag{A1-7}$$

The non-negativity constrained distortion, $D'(\frac{1}{2})$,

$$D'(\tfrac{1}{2}) = \sigma_1^2 2^0 + \sigma_2^2 2^{-2} = \tfrac{1}{4} + 1 \tag{A1-8}$$

is greater than $D(\frac{1}{2})$.

By forcing the rates to be greater that 0, the distortion level suffers. But, when adding additional constraints as we have done here, the set of admissible rates is smaller and the a change in the distortion value results.

Appendix 2.

Another Approach to the Bit Allocation Problem

Reconsider the solution for the optimal coefficient rates assigns for the distortion function of (6-16) and (6-17). That is, minimize the distortion

$$\min_{\{B_i\}} D = \sum_{i=1}^{n} \epsilon_i^2 \sigma_i^2 2^{-2B_i} \tag{A2-1}$$

subject to

$$\sum_{i=1}^{n} B_i = nB, \text{ and } B_i \geq 0 \ \forall i. \tag{A2-2a,b}$$

As was indicated in Chapter 6, the optimal rates are assigned through the solution of n partial differential equations

$$\frac{\partial D}{\partial B_i} = (-2\ln 2)\epsilon_i^2 \sigma_i^2 2^{-2B_i} + \frac{\partial \epsilon_i^2}{\partial B_i} \sigma_i^2 2^{-2B_i} \ \ \forall i \tag{A2-3}$$

Rewrite the i-th equation as an equation of differentials

$$\left\{\sigma_i^2 2^{-2B_i}\right\} d\epsilon_i^2 + \left\{\lambda - (2\ln 2)\epsilon_i^2 \sigma_i^2 2^{-2B_i}\right\} dB_i = 0 \tag{A2-4}$$

This equation can be represented more simply as

$$M(\epsilon_i^2, B_i) d\epsilon_i^2 + N(\epsilon_i^2, B_i) dB_i = 0 \tag{A2-5}$$

Since $\frac{\partial N}{\partial \epsilon_i^2} = \frac{\partial M}{\partial B_i}$, equation (5) is an "exact" partial differential equation [79]. Its solution, $F(\epsilon_i^2, B_i)$, is known to have the property [79]

$$\frac{\partial F}{\partial \epsilon_i^2} d\epsilon_i^2 + \frac{\partial F}{\partial B_i} dB_i = 0 \tag{A2-6}$$

This means that F is constant function in both the performance factor and the coefficient rate

$$F(\epsilon_i^2, B_i) = c \tag{A2-7}$$

The value of the constant, c, can be obtained from a set of boundary conditions on F. More is said about this later.

$F(\epsilon_i^2, B_i)$ can be found by several methods. The "standard method" involves solving for F by integrating one of the partial derivatives of (6), say $\frac{\partial F}{\partial \epsilon_i^2}$; this leads to

$$F(\epsilon_i^2,B_i) = \int M\partial\epsilon_i^2 + \phi(B_i) = \epsilon_i^2\sigma_i^2 2^{-2B_i} + \phi(B_i) \tag{A2-8}$$

The integration factor, $\phi(B_i)$, can be found by using the other partial derivative, $\frac{\partial F}{\partial B_i}$=N,

$$\frac{\partial F}{\partial B_i} = -(2\ln 2)\epsilon_i^2\sigma_i^2 2^{-2B_i} + \frac{\partial\phi(B_i)}{\partial B_i} = \lambda - (2\ln 2)\epsilon_i^2\sigma_i^2 2^{-2B_i} \tag{A2-9}$$

By collecting terms of this equation, we see that

$$\frac{\partial\phi(B_i)}{\partial B_i} = \lambda \tag{A2-10}$$

Integrating (10) with respect to B_i shows that $\phi(B_i)$ is linearly proportional to coefficient rate

$$\phi(B_i) = \lambda B_i + c_0 \tag{A2-11}$$

where c_0 is another constant of integration. By combining this with (7), F can be found

$$F = \epsilon_i^2\sigma_i^2 2^{-2B_i} + \lambda B_i + c_0 = c \tag{A2-12}$$

Combining the two constants, as $k=c_0-c$, F is described by a family of curves

$$F(\epsilon_i^2,B_i) = \epsilon_i^2\sigma_i^2 2^{-2B_i} + \lambda B_i + k = 0 \tag{A2-13}$$

The constant k can be found from boundary conditions: ϵ_i^2=1 at B_i=0 (see Figure 6.2)

$$F(1,0) = \epsilon_i^2\sigma_i^2 + k = 0 \Rightarrow k = -\sigma_i^2 \tag{A2-14}$$

Therefore, the solution to the i-th PDE is the solution of an algebraic equation

$$F_i \equiv F = \epsilon_i^2\sigma_i^2 2^{-2B_i} + \lambda B_i - \sigma_i^2 = 0 \tag{A2-15}$$

The Lagrange constant, λ, can be found using its associated constraint (2b) in the sum

$$\sum_{j=1}^{n} F_i = \sum_{j=1}^{n} \epsilon_i^2\sigma_i^2 2^{-2B_i} + \lambda nB - \sum_{j=1}^{n} \sigma_i^2 = 0 \tag{A2-16}$$

Substituting this into (15) reveals that the n coefficient rates are the solutions of a set of n transcendental equations

$$B_i = \frac{nB\left(\sigma_i^2 - \epsilon_i^2\sigma_i^2 2^{-2B_i}\right)}{\sum_{j=1}^{n}\sigma_j^2 - \sum_{j=1}^{n}\epsilon_j^2\sigma_j^2 2^{-2B_i}} \quad \forall i \tag{A2-17}$$

The notation of this equation can be simplified by noticing that the denominator is determined from the distortion function (1) evaluated at two different rates

$$D(0) = \sum_{j=1}^{n} D_i(0) = \sum_{j=1}^{n} \sigma_i^2 \text{ and } D(B) = \sum_{j=1}^{n} D_i(B_i) = \sum_{j=1}^{n} \epsilon_i^2\sigma_i^2 2^{-2B_i} \tag{A2-18}$$

Thus, the coefficient rates become

$$B_i = nB\left\{\frac{D_i(0) - D_i(B_i)}{D(0)-D(B)}\right\} \quad \forall i \tag{A2-19}$$

Consider what happens as $B \rightarrow \infty$. The solution of this appendix goes to infinity,

$$B_i \rightarrow \left\{\frac{n\sigma_i^2}{\sum_{j=1}^{n}\sigma_j^2}\right\}B = \left\{\frac{nD_i(0)}{D(0)}\right\}B \rightarrow \infty \text{ as } B \rightarrow \infty \tag{A2-20}$$

but it does so differently than the solutions found in Chapter 6. In Chapter 6, the coefficient rates approach infinity with unit slope. Here each coefficient approaches infinity with a different slope. Why this happens is still open for further consideration. The difference between these two solutions may be a result of the non-guaranteed uniqueness of the relationship between λ and B_i in (15).

References

[1] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prenctice-Hall, 1984.

[2] P. Hecbert, "Color image quantization for frame buffer display," *ACM Computer Graphics*, vol. 16, pp. 297-307, July 1982.

[3] B. R. Hunt, "Nonstationary statistical image models (and their application to image data compression)," *Comput. Graphics, Image Processing*, pp. 173-186, 1980.

[4] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1258-1268, Oct. 1986.

[5] J. L. Mannos amd D. L. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 525-536, July 1974.

[6] CCIR Recommendation 601, "Encoding parameters of digital television for studies," CCIR Recomm. and Reports, vol. XI, ITU, Geneva, Switzerland, 1982.

[7] W. K. Pratt, "Spatial Transform Coding of Color Images," *IEEE Trans. Commun.*, vol. COM-51, pp. 980-992, Dec. 1971.

[8] W. N. Sproson, *Colour Science in Television and Display Systems*, Philadelphia: Heyden and Sons, 1983.

[9] R. K. Jurgen, "High-definition television update," *IEEE Spectrum*, vol. 25, no. 4, pp. 56-62, April 1988.

[10] E. E. Hilbert, "Cluster compression algorithm: A joint clustering/data compression concept," Jet Propulsion Laboratory, Pasedena, CA, Publ. 77-43.

[11] E. E. Hilbert, "Joint pattern/recognition/data compression concepts for ERTS multispectral data," SPIE, vol. 66, "Efficient transmission of pictoral information," Aug. 1975.

[12] W. D. Hoffman and D. E. Troxel, "Making Progressive Transmission Adaptive," *IEEE Trans. Commun.*, vol COM-34, pp. 806-813, Aug. 1986.

[13] F. Kretz, "Subjectively optimal quantization of pictures," *IEEE Trans. Commun.*, vol. COM-23, pp. 1288-1292, Nov. 1975.

[14] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, N0. 3, pp. 349-289, March 1981.

[15] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366-406, March 1980.

[16] J. O. Limb, C. B. Rubinstein and J. E. Thompsom, "Digital coding of color video—A review," *IEEE Trans. Commun.*, vol. COM-25, pp. 1349-1385, Nov. 1977.

[17] W. K. Pratt, "Spatial Transform Coding of Color Images," *IEEE Trans. Comm.*, vol. COM-51, pp. 980-992, Dec. 1971.

[18] W. K. Pratt, "Spatial Transform coding of color images," *IEEE Trans, Commun. Technol.*, vol. COM-19, pp. 980-992, Dec. 1971.

[19] J. D. Eggerton and M. D. Srinath, "A visually weighted quantization for images bandwidth compression at low data rates," *IEEE Trans. Commun.*, vol. COM-34, pp. 840-847, Aug. 1986.

[20] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1258-1268, Oct. 1986.

[21] B. Girod, H. Almer, L. Bengtsson, B. Christensson and P. Weiss, "A subjective evaluation of noise-shaping quantization for adaptive intra-/interframe DPCM coding of color television signals," *IEEE Trans. Commun.*, Vol. COM-36, no. 3, pp. 332-346, March 1988.

[22] J. J. Koenderink, W. A. van de Grind and M. A. Bouman, "Foveal information processing at photopic luminances," *Kybernetik*, vol. 8, no. 4, pp/ 128-144, 1971.

[23] F. X. J. Lukas and Z. L. Budrikis, "Picture quality prediction based on a visual model," *IEEE Trans. Commun.*, vol. COM-30, pp. 1679-1692, July 1982.

[24] W. K. Pratt, *Digital Image Processing*, New York, NY: Wiley, 1978.

[25] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1445-1453, Sep. 1988.

[26] K. K. Paliwal and V. Ramasubramanian, "Effect of ordering the codebook on the effciency of the partial distance search algorithm for vector quantization," to appear.

[27] J. D. Gibson and K. Sayood, "Lattice quantization," to appear in *Advances in Electronics and Electron Physics.*

[28] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 337-380, July 1979.

[29] Y. Yamada, S. Tazaki and R. M. Gray, "Asymptotic performance of a block quantizer with difference distortion measure," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 6-14, Jan. 1980.

[30] J. A. Bucklew and G. L. Wise, "Multidimensional asymptotic quantization theory with rth power distortion measures," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 238-247, Mar. 1982.

[31] J. A. Bucklew and G. L. Wise, "Companding and random quantization in several dimensions," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 207-211.

[32] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211-226, Mar. 1982.

[33] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227-232, Mar. 1982.

[34] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29 pp. 820-824, Nov. 1983.

[35] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[36] W. H. Equitz, "Fast algorithms for vector quantization picture coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1987, pp. 725-728.

[37] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568-583, July 1896.

[38] S. Yuan and K.-B. Yu, "Zonal sampling and bit allocation of HT coefficients in image data compression," *IEEE Trans. Commun.*, vol. COM-34, pp. 1246-1251, Dec. 1986.

[39] N. Ahmed, T. Natarajan and K. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan. 1974.

[40] W. H. Chen and C. H. Smith, "Adaptive coding of color images," *IEEE Trans. Commun.*,

vol. COM-25, pp. 1285-1292, Nov. 1977.

[41] C. E. Shannon, "A mathematical theory of communications," *Bell System Tech. Jour.*, vol. 27, pp. 379-423, July 1948.

[42] D. A. Huffman, "A method for the construction of minimum-redundant codes," *Proc. IRE*, vol. 40, pp. 1098-1101, Sept. 1952.

[43] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, New York: Springer-Verlag, 1975.

[44] H.-M. Hang and B. G. Haskell, "Interpolating vector quantization of color images," *IEEE Trans . Commun.*, vol. 36, pp. 465-470, April 1988.

[45] D. H. Halverson, N. C. Griswold and G. L. Wise, "A generalized block truncation algorithm for image compression," *IEEE Trans. Acoust. Speech Signal Proc.*, vol. ASSP-32, June 1984.

[46] L. Wang and M. Golderg, "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. COM-36, pp. 79-87, Jan. 1988.

[47] S. L. Tanimoto, "Image transmission with gross information first," *Comput. Graphics Image Processing*, vol. 9, pp. 72-76, Jan. 1979.

[48] K. Sloan and S. L. Tanimoto, "Progressive refinement of raster images," *IEEE Trans. Comput.*, vol. C-28, pp. 871-874, Nov. 1979.

[48] K. Knowlton, "Progressive transmission of grey scale and binary pictures by simple, efficient, and lossloss encoding scheme," *Proc. IEEE*, vol. 68, pp. 885-896, July 1980.

[50] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532-540, Apr. 1983.

[51] T. Kronander, "Sampling of bandpass pyramids," *IEEE Trans. Commun.*, vol. COM-36, pp. 125-127, Jan. 1987.

[52] E. H. Adelson and P. J. Burt, "Image data compression with the Laplacian pyramid," *Proc. IEEE Conf. Pattern Recognition Image Processing*, pp. 218-223, 1981.

[53] K. N. Ngan, "Image display techniques using the cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 173-177, Feb. 1984.

[54] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[55] W. K. Pratt, *Image Transmission Techniques*, New York: Academic, 1979.

[56] W. K. Pratt, *Digital Image Processing*, New York: Academic, 1978.

[57] E. Dubois and J. L. Moncet, "Encoding and progressive transmission of still pictures in NTSC composite format using transform domain methods," *IEEE Trans. Commun.*, vol. COM-34, pp. 310-319, Mar. 1986.

[58] W. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, pp. 225-232, Mar. 1984

[59] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 162-169, Mar. 1976.

[60] H. M. Dreizen, "Content-driven progressive transmissiom of grey-scale images," *IEEE Trans. Commun.*, vol. COM-35, pp. 289-296, Mar. 1987.

[61] K. R. Sloan, Jr. and S. L. Tanimoto, "Progressive refinement of raster scan images," *IEEE*

Trans. Comput., vol. C-28, pp. 871-874, Nov. 1979.

[62] A. J. Frank, J. D. Daniels and D. R. Unangst, "Progressive image transmission using a growth geometry coding," *Proc. IEEE*, vol. 68, pp. 897-909, July 1980.

[63] K. Knowlton, "Progressive transmission of grey-scale and binary image by simple, efficient, and lossless encoding schemes," *Proc. IEEE*, vol. 68, pp. 885-896, July 1980.

[54] E. J. Delp and O. J. Mitchell, "Image truncation using block truncation coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 1335-1342, Sept. 1979.

[65] W. D. Hoffman and D. E. Troxel, "Making Progressive Transmission Adaptive," *IEEE Trans. Commun.*, vol. COM-34, pp. 806-813, Aug. 1986.

[66] D. J. Vaisey and A. Gersho, "Variable block-size image coding," *Proc. ICASSP*, pp. 1051-1054, Apr. 1987.

[67] A. J. Frank, J. D. Daniels and D. R. Unangst, "Progressive image transmission using a growth-geometry coding," Proc. IEEE, vol. 68, pp. 898-909, July 1980.

[68] H. Samet, "The quad tree and related hierarchical data structures," ACM Computing Surveys, vol. 16, no. 2, June 1984.

[69] M. Kunt, et.al. "Second generation image coding techniques," Proc. IEEE, vol. 73, pp.549-574, Apr. 1985.

[70] J. J. Y. Huang and P. M. Schultheiss, "Block quantization of correlated random variables," *IEEE Trans. Commun. Syst.*, vol CS-11, pp. 289-296, Sept. 1963.

[71] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6. pp. 7-12, March 1960.

[72] L. Wang and M. Goldberg, "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. COM-36, no. 1, pp. 75-87, Jan. 1988.

[73] W. C. Adams and C. E. Giesler, "Quantizing characteristics for signals have laplacian amplitude probability density function," *IEEE Trans. Commun.*, vol. COM-26, no. 8, Aug. 1978.

[74] T. Berger, *Rate Distortion Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1971.

[75] M. D. Paez and T. H. Glisson, "Minimum mean square error quantization in speech PCM and DPCM systems," *IEEE Trans. Commun.*, pp. 225-230, April 1972.

[76] D. G. Luenberger,"Linear and Nonlinear Programming," Reading, Massachusetts: Addison-Wesley, 1984.

[77] R. C. Reininger and J. D. Gibson, "Distribution of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 835-839, June 1983.

[78] D. G. Luenberger, *Optimization by Vector Space Methods*, New York, NY: Wiley, 1969.

[79] S. L. Ross, "Introduction to Ordinary Differential Equations", Lexington, Massachusetts: Xerox College Publishing, 1974.

[80] E. N. Gilbert, "Synchronization of binary messeges," *IRE Trans. Inform. Theory*, pp. 470-477, Sept. 1960.

[81] P. Noll and R. Zenlinski, "Bounds on quantizer performance in the low bit-rate region," *IEEE Trans. Comm.*, vol. COM-26, pp. 300-304, Feb. 1978.